Mesurer la confidentialité avec des métriques de discernabilité: définitions, mécanismes et confidentialité des informations liées à la localisation

Nicolás E. Bordenabe

École Polytechnique
Thèse de Doctorat
Spécialité Informatique

Measuring Privacy with Distinguishability Metrics: Definitions, Mechanisms and Application to Location Privacy

Présentée et soutenue publiquement par

Nicolás E. Bordenabe

le 12 septembre 2014

devant le jury composé de

Rapporteurs:	Gilles Barthe
	George Danezis
Directeur de thèse:	Catuscia Palamidessi
	Konstantinos Chatzikokolakis
Examinateurs:	Daniel Augot
	Pedro R. D'Argenio
	Reza Shokri

Abstract

The increasing availability of smartphone and tablets has given place to the development of a broad new class of applications, which collect and analyze big amounts of information about its users for different reasons: offering a personalized service, offer targeted advertisement, or provide accurate aggregated data for research and analysis purposes. However, serious privacy concerns have been risen about the kind and quantity of data being collected: this data is in general private by nature, and often it can be linked to other kinds of sensitive information. And in most cases, this information is made available to an untrusted entity, either because the service provider itself is not reliable, or because some aggregated information is being publicly released. In order to deal with these concerns, some kind of privacy guarantee is needed. Differential Privacy is one of the most prominent frameworks used to deal with disclosure prevention in statistical databases. It provides a formal privacy guarantee, ensuring that sensitive information relative to individuals cannot be easily inferred by disclosing answers to aggregate queries. If two databases are adjacent, i.e. differ only for an individual, then the query should not allow to tell them apart by more than a certain factor. This induces a bound also on the distinguishability of two generic databases, which is determined by their distance on the Hamming graph of the adjacency relation. When the sensitive information to be protected is other than the value of a single individual, or when the secrets itself are not databases at all, it is common to consider different notions of distinguishability, which depend on the application at hand and the privacy guarantees we wish to express.

In the first part of this thesis we explore the implications of differential privacy when the indistinguishability requirement depends on an arbitrary notion of distance. We show that we can naturally express, in this way, (protection against) privacy threats that cannot be represented with the standard notion, leading to new applications of the differential privacy framework. We give intuitive characterizations of these threats in terms of Bayesian adversaries. We revisit the well-known results about universally optimal mechanisms, and show that, in our setting, these mechanisms exist for sum, average, and percentile queries.

In the second part of this thesis we introduce *geo-indistinguishability*, a formal notion of privacy for location-based systems. This privacy definition corresponds to an instance of the generalized version of differential privacy presented before. We also show a mechanism for achieving this notion and study different issues that arise with its implementation, namely the truncation of the result and the effect of the precision of the machine. We also describe how to use our mechanism to enhance LBS applications with geo-indistinguishability guarantees without compromising the quality of the results.

In the last part of this thesis, we consider the location privacy framework of Shokri et al., which offers an optimal trade-off between the loss of quality of service

and the privacy protection with respect to a given Bayesian adversary. We show that it is possible to combine the advantages of this approach with ours: given a minimum threshold for the degree of geo-indistinguishability, we construct a mechanism that offer maximal utility, as the solution of a linear optimization problem. Since geo-indistinguishability is insensitive to the remapping of a Bayesian adversary, this mechanism is optimal also in the sense of Shokri et al. Furthermore we propose a method to reduce the number of constraints of the linear program from cubic to quadratic, enlarging significantly the size of location sets for which the optimal trade-off mechanisms can still be computed, while maintaining the privacy guarantees without affecting significantly the utility of the generated mechanism.

Résumé

La disponibilité croissante de smartphones et tablettes a donné lieu à l'élaboration d'une vaste classe de nouvelles applications, qui recueillent et analysent de grandes quantités d'informations sur leurs utilisateurs pour des raisons différentes: offrir un service personnalisé, offrir de la publicité ciblée, etc. Toutefois, le type et la quantité de données collectées ont engendres des graves préoccupations concernant la vie privée: en effet, ces données sont en général confidentielles par nature, et souvent, elles peuvent être liées à d'autres types d'informations sensibles. Afin de pallier à ces préoccupations, des garanties de confidentialité sont nécessaires. Differential privacy est l'une des notions de confidentialité les plus importantes dans le contexte des bases de données statistiques. Elle fournit une garantie formelle de confidentialité, assurant qu'aucune information sensible concernant des particuliers ne peut être facilement déduite par la divulgation des réponses aux questions globales. Si deux bases de données sont adjacentes, c'est à dire ne diffèrent que pour un individu, la requête ne devrait pas permettre de les distinguer par plus d'un certain facteur. Ceci induit une borne sur la discernabilité qui est déterminée par la distance sur le graphe de Hamming de la relation de contiguïté. Lorsque les informations sensibles à protéger ne sont pas les données relatives à un seul individu, ou lorsque les secrets se sont pas du tout les bases de données, il est courant de considérer les différentes notions de discernabilité, qui dépendent de l'application et de la garantie de confidentialité que nous voulons exprimer.

Dans la première partie de cette thèse, nous explorons les implications de la differential privacy lorsque l'exigence d'indiscernabilité repose sur une notion arbitraire de la distance. Nous pouvons exprimer de cette façon les menaces contre la vie privée qui ne peuvent pas être représentées par la notion standard. Nous donnons des caractérisations intuitives de ces menaces en termes d'adversaires bayésiens. Nous revisitons les résultats connus sur les mécanismes universellement optimaux, et nous montrons que, dans notre contexte, ces mécanismes existent pour les requêtes somme, moyenne, et percentile .

Dans la deuxième partie de cette thèse, nous introduisons le concept de géo-indiscernabilité, une notion formelle de confidentialité pour les systèmes basés sur la localisation. Cette définition est un cas particuliere de la version généralisée de la differential privacy présenté precedemment. Nous présentons aussi un mécanisme qui permet d'atteindre cette notion et nous étudions les différentes questions que pose la mise en œuvre, à savoir la troncature du résultat et l'effet de la précision de la machine. Nous décrivons également comment utiliser notre mécanisme pour améliorer les applications LBS avec des garanties de géo-indiscernabilité sans compromettre la qualité des résultats.

Dans la dernière partie de cette thèse, nous considérons le méchanisme de Shokri et al, qui offre un compromis optimal entre la perte de qualité de service et la protection de la vie privée par rapport à un adversaire bayésien. Nous montrons qu'il est

possible de combiner les avantages de cette approche avec la nôtre: étant donné un seuil minimal pour le degré de géo-indiscernabilité, nous construisons un mécanisme qui offre utilité maximale, en resolvant un problème d'optimisation linéaire. Puisque la géo-indiscernabilité est insensible à la reconfiguration d'un adversaire bayésien, ce mécanisme est également optimal dans le sens de Shokri et al. En outre, nous proposons une méthode pour réduire le nombre de contraintes du programme linéaire de cubique è quadratique, élargissant considérablement la taille des ensembles de localisations pour lesquels les mécanismes optimaux peuvent encore être calculés, tout en maintenant les garanties de confidentialité sans affecter significativement l'utilité du mécanisme généré.

Acknowledgements

First of all, I want to thank my supervisors, Catuscia Palamidessi and Konstantinos Chatzikokolakis, for their constant support and dedication over the past three years, which have been the cornerstone of this work. From the beginning they have provided a welcoming and nourishing environment, which turned these years into one of the best experiences of my life. They are not only the brightest persons I know, but also wonderful human beings with whom I shared innumerable good moments, lots of enlighting discussions, and even some challenging all-night-up sessions before a deadline. Working with them during this time has been an honour and a pleasure.

I also want to thank INRIA (Institut National de Recherche en Informatique et en Automatique) and the DGA (Direction Générale de l'Armement) for providing the funds for the duration of my PhD.

I am also deeply thankful to the *rapporteurs* of my thesis, Gilles Barthe and George Danezis, for taking part of their invaluable time to evaluate this work, and also to the rest of the members of the jury, Reza Shokri, Pedro D'Argenio and Daniel Augot, for reading it and providing valuable feedback.

I would also like to thank the wonderful people in the Comète team that, one way or another, helped making this whole experience unforgettable. I will be always grateful to Miguel Andrés, who was my first contact for this opportunity and my collegue and coauthor during my first year in the team. I consider him a great career advisor and a wonderful friend. I am also thankful to Luis Pino who was not only a friend but a brother to me during this three years, and with whom I shared more good moments than I could possibly count, and to his girlfriend Claudia Scioli, for being present in most of them. I am deeply thankful to Frank Valencia, for his invaluable frienship, his immeasurable help from the very beginning of my stay in France, and for allowing me to "borrow" his piano for almost two years. I'm also greatful to Marco Stronati, for being a great friend during these three years. I would also like to thank Andrés Artistizabal, Lili Xu, Salim Perchy, Sophia Knight, Ehab El-Salamouny, Sardouna Hamadou, Matteo Cimini, Yusuke Kawamoto and Thomas Given-Wilson, for all the shared moments, the poker nights and for making Comète one of the nicest groups I have ever been.

Finally, I would like to thank my wife, Andrea Mairone, for sharing this adventure with me, for all the trips and good moments spent together during this time, and for making these last three years so much enjoyable. And I would also like to thank my family, for their constant support and affection.

Contents

Chapter 1

Introduction

1.1 Motivation

The fact that individuals willingly share personal information is not new. A common example is the case of surveys and censuses, collecting different kinds of personal data about people, with various and different purposes. However, over the last few years there has been a pronounced increase in the amount of information that individuals share, as well as in the ways this information is shared. One of the main reasons of this phenomenon is the broad adoption of hand-held devices such as smartphones and tablets, combined with the high availability of wireless internet connection. Nowadays, people are connected to the Internet (actively or not) at almost every instant of their lives. This, in turn, has motivated the development of a whole new range of services (like social networks, photo sharing applications, etc.) that allow an individual to share various kinds of information with friends or even with strangers. Other applications rely on some particular type of information being shared by the user (e.g. his location, his interests, etc) to provide a requested service (e.g. locate restaurants around the user's current location, provide a personalised news feed, etc). Recent studies in the US show that in 2013, 56% of the adult population of the country owns a smartphone (in comparison with 35% in 2011) [Pew Internet. Smartphone Ownership 2013]. Of these users, 40% use some kind of social network in their phone (28% do so in a daily basis), and 74% use services based on their location [Pew Internet. Location-Based Services 2013].

However, it is worth noting that the data being shared is, in general, collected and analysed by the service provider. Moreover, other types of services like e-commerce portals, search engines and email providers also collect and process data that, in principle *are supposed to remain private* (like browsing history, messages contents, etc). Besides, by studying and processing this data, the service provider is able to learn even new information about their users (e.g. browsing habits, personal interests, sexual orientation, etc.) that was never explicitly provided by them. In some cases, it could even be said that the service provider knows things about their users that the users themselves do not know. The provider can use this information with various purposes: enhancing the service itself, predicting the behaviour and interests of users, offering targeted advertisement, etc. But in other cases, for instance when the service provider is not trustworthy, personal data can also be used with various malicious intentions: monitoring persons, fraud, scam and discrimination.

This clearly implies that measures need to be taken in order to protect the privacy of those individuals that share information. From the point of view of a

user, it is important that he knows what information can be learned (explicitly or implicitly) by the provider, which parts of this information are disclosed, how his data is being used, and how is his privacy affected by the combination of these factors. But at the same time, from the point of view of the provider, it is important that users feel comfortable with the service and the privacy guarantees it offers, so that they keep using it.

In order to design mechanisms for privacy protection, it is key to have a definition of privacy. However, we note that defining privacy is not easy: on the one hand, the definition could be tied to the context in which it is being used (the definition of privacy in the context databases might differ, for instance, from the definition in the context of geolocation applications); and on the other hand, there might be different definitions that depend on what the user is interested to protect (e.g his identity, the content of the data, the accuracy with which his information can be inferred, etc). Our goal is to have a general definition of privacy that can capture a broad range of scenarios. We also aim at a *formal* notion of privacy, since it is important that we can verify if a mechanism satisfies this property or not. Moreover, a formal notion will in general allows us to quantify the level of privacy of a given mechanism, which will be useful to evaluate and compare different mechanisms.

1.1.1 Defining and measuring privacy

The literature is rich in works attempting to define privacy and offering means to measure it. We note, however, that in general the way of defining privacy depends heavily on the nature of the scenario in which it is being applied. Some privacy guarantees for location applications might not be suitable for, say, databases. In this section, we recall some of the most important works that attempt to define and quantify privacy.

In principle, when defining privacy, we can broadly distinguish two different kinds of approaches: those attempting to protect the *user's identity*, and those focused on protecting the *user's data*. In the former, privacy is achieved by preventing an adversary from linking the data corresponding to an individual (which is disclosed to the provider) with the individual's identity. In the latter, on the other hand, the identity of the data's owner is assumed to be known by the adversary. Privacy is then obtained by modifying the information disclosed to adversary, for instance by replacing it with an approximate value, increasing its granularity, adding dummy results, etc.

Protecting user's identity

One of the most prominents approaches to protect the identity of an individual is the notion of k-anonymity. This concept was first introduced in [Samarati 2001] in the context of statistical databases. When selecting a set of records from the database to be released, the definition of k-anonymity requires that each combination of "quasi-identifiers" (that is, set of attributes that, combined, can be used to

identify an individual) appearing in the released set is repeated in at least k records. The intuitive idea is that an adversary should not be able to distinguish the data belonging to an individual from, at least, another $k-1$ individuals. To achieve this, some attributes can be discretized (by reducing the granularity of the set of possible values) or simply hidden. Several variants of this notion have been used in contexts different than the usual case of statistical databases, for instance to define privacy in location-based systems [Gruteser 2003, Gedik 2005, Mokbel 2006]. Also, extensions of this concept have been introduced in order to cope with the weaknesses of the approach. The l-diversity notion [Machanavajjhala 2007] requires that, for each combination of non-sensitive attributes present in the released set, there are at least l "well represented" values for each sensitive attribute (this could be, for instance, l different values). The t-closeness notion [Li 2007] asks that the difference between the distribution of the values of a sensitive attribute in an equivalence class and the distribution in the whole database is bound by a threshold t.

However, due to the increasing amount of information about individuals being publicly available, several studies show that, in some cases, protecting the user's identity might not be enough, since an attacker might be able to link the disclosed data with their owners with high precision. In [Narayanan 2009] the authors present a de-anonymization algorithms for social-network graphs that allows them to re-identify a third of the users having accounts on both Twitter and Flickr with a low error rate. In a different work [Narayanan 2008], the same authors apply a de-anonymization algorithm to the Netflix Prize Dataset (containing movie ratings information of 500,000 users) to link anonymous records to known Netflix users, by using the auxiliary information provided by IMDB.com. This way, they were able to infer other types of private information (like political and religious views) of the compromised users. The deanonymization threat is even more evident in the context of location privacy: the location of an individual has not much value by itself, but in general there is a huge amount of other information that that can be inferred from it, like his work and home address, his movement patterns, and his hobbies [Gambs 2011].

Also, it is important to note that, in some contexts, hiding the user's true identity might not an option. Suppose for instance an application that provides the user with personalized contents, like a news feed based on his interests, a list of restaurants nearby based on his location, or a music recommendation platform based on the what the user's friends are listening. In most of these applications the user is assumed to be authenticated in the service, providing information like name and email address, and therefore anonymity is lost. This scenario is the one we assume in the rest of this thesis.

Protecting user's data

Techniques that aim at protecting the user's data are generally based on disclosing a modified version of the information to the adversary in order to reduce his accuracy. Most of these techniques can be combined with the techniques mentioned before,

but more importantly, they can be used in the case the service provider is assumed to know the identity of the user (for instance, if the user is authenticated in the application).

The way to measure privacy for these kind of techniques depends, in general, on the scenario being considered. In the context of location-based systems, for instance, one of the most common notions used to quantify privacy is the expected error of the adversary trying to guess the real location of an individual from the reported one. In the same context, we can find privacy definitions based on variations of differential privacy or k-anonymity (based on cloaking or dummy locations), although these last ones have been found to be weak in such scenario [Shokri 2010] . These notions, as well as some works in the literature based on them, will be discussed in more detail in Chapter 3.

The concept of differential privacy, introduced in [Dwork 2006a], is one of the most important privacy notions for obfuscation techniques used in the context of statistical databases. In a nutshell, differential privacy requires that the observations reported from "similar" databases (that is, databases differing in only one record) should be generated with similar probabilities. An important property of this notion is that its definition does not make any kind of assumption about the prior knowledge of the adversary.

In [Ghosh 2009] the authors provide a mechanism for counting queries, which is a discrete variant of the well known Laplace mechanism, that achieves optimal utility for a fixed level of differential privacy, any user and any prior information available to the attacker. The authors consider the (inverse of the) bayesian notion of utility, which measures the expected loss between the real answer of the query and the reported result. The authors of [Gupte 2010] prove the same optimality result for the case of minimax utility, that measures the maximum expected loss for any query result. Finally, in [Brenner 2010] the authors show that these optimal mechanisms only exist for counting queries.

Differential privacy has been used in contexts other than statistical databases. It has, for instance, gained a lot of attention in the context of smart metering, since reassuring users with strong privacy guarantees is key for their deployment. In [Danezis 2011] the authors use the differential privacy framework to enhance fine-grained billing (like smart metering for electricity, on-demand TV content, etc) with privacy guarantees. The basic idea is that a small amount of noise should be added to each individual payment or reading, in a way to ensure differential privacy guarantees on the total amount to be payed. However, since this would lead the user to pay more than he should, a cryptographic protocol is proposed to future track and reclaim the extra amount payed due to privacy protection. The authors of [Ács 2011] rely on adding noise to the individual readings in such a way that the total aggregated noised reading for a cluster (a set of users or households) is as accurate as possible. Also in the case of location-based systems, several authors have proposed differential privacy based approaches to protect users' privacy. These works will be discussed in more detail in Section 3.2.

There are other works that attempt to generalize the notion of differential privacy by extending the metric used to measure the distance between databases. The authors of [Reed 2010] introduce the idea of a general metric for differential privacy, and develop a type system that can be used to model algorithms that offer differential privacy in the standard way. In [McSherry 2007] the authors design a generalized version of the Laplace mechanism that adds noise to non-numerical query results, by means of a scoring function. In practice, however, no metric other than the standard hamming distance is used in any of the previously mentioned works.

1.1.2 Goals

In this work, we are interested in deriving a formal notion of privacy that can be used measure and quantify the privacy of a given application. We will focus on the case where the information to be protected is the user's data and not the user's identity. As we mentioned before, one of the most prominents notions for this scenario is differential privacy. However, and although some generalized versions of this notion exist in the literature, no definition other than the standard one have been used in practice. We are interested in a notion that can be applied to an arbitrary domain of secrets, in which there might be no information aggregation at all.

We set as the main goal of this thesis the study and evaluation of a privacy framework, based on a generalized notion of differential privacy, that is suitable for arbitrary domains of secrets. We argue that this can be done by paying particular attention to the notion of distance between secrets (corresponding to the adjacency relation between databases in standard differential privacy). For an arbitrary set of secrets, the distance function can be interpreted as the level of distinguishability between secrets. Therefore, by defining and manipulating this distance, we would be able to express different scenarios and privacy threats, and to develop corresponding privacy protection mechanisms for each case.

As a running example, we consider a user located in Paris who wishes to use a location-based service (that is, an application that provides a service based on the current location of the user) to find nearby restaurants in a private way. In principle, this could be achieved by disclosing some approximate information z instead of his exact location x. However, in order to get any useful privacy guarantee, this approximate information cannot be generated naively. It is clear that there is no direct way to apply the standard notion of differential privacy in this context: there is no database involved, and no information to be aggregated. In the rest of this thesis, we will aim at developing a generalized version of the definition of differential privacy that can be successfully applied in this scenario. We will also design mechanism that can be used to achieve this notion, and evaluate them in terms of utility, privacy and complexity. However, these mechanisms will be designed in such a way that they can be implemented directly into the user's device, without modifying the service provider, since this will allow us to use them within any existing location-aware application. Moreover, in some cases this might be the only possible solution, since

in general providers do not have enough incentives to modify their services in order to add privacy guarantees.

1.2 Plan of the thesis and contributions

In this section we present a brief description of the content of each of the following chapters, as well as the contributions in each of them.

In Chapter 2 we review some basic notions and results that will be used in the rest of the thesis, including concepts and basic results on metrics, probabilities, and mechanisms. We also devote a small section to the basics of standard differential privacy.

In Chapter 3 we present the state-of-the-art on location privacy metrics. We review in detail two notions to measure the privacy and utility of a location obfuscation mechanism, that will be widely used in later chapters. We also present a mechanism, based on these two notions, that achieve optimal privacy guarantees under certain conditions. Finally, we review various different ways to measure location privacy, addressing the strengths and weaknesses of each. This will help us defining the specific goals to be pursued when building our own notion of location privacy.

In Chapter 4 we present a generalization of the popular notion of differential privacy. This generalization assumes a generic domain of secrets equipped with a *privacy metric*. This way, the definition can be adapted to numerous scenarios where the standard definition would not be convenient (or even possible) to apply. We present intuitive characterizations of the different privacy threats in terms of a Bayesian adversary, which generalize the two most common interpretations of the standard definition of differential privacy. We revisit the well-known results stating that universally optimal mechanisms exist only for counting queries: we show that, in our extended setting, universally optimal mechanisms exist for other queries too, notably sum, average, and percentile queries. Finally, we explore various applications of the generalized definition, for statistical databases as well as for other areas, such as smart metering.

In Chapter 5 we introduce geo-indistinguishability, a novel definition of location privacy, derived as an instance of the previous generalized privacy definition by considering the set of secrets to be a set of spatial locations. This privacy definition formalizes the intuitive notion of protecting the user's location within a radius r with a level of privacy that depends on r. Furthermore, we present a mechanism for achieving geo-indistinguishability by adding controlled random noise to the user's location. We describe how to use our mechanism to enhance location-based service (LBS) applications with geo-indistinguishability guarantees without compromising the quality of the application results. We also show, through a case study, how the proposed method can be used to sanitise a dataset containing location data without heavily affecting the obtained results. Finally, we compare state-of-the-art

mechanisms from the literature with ours. We will see that, among all mechanisms independent of the prior, our mechanism offers the best privacy guarantees.

In Chapter 6 we tackle the problem of the trade-off between privacy and utility of a geo-indistinguishable mechanism. We show that, given a desired degree of geo-indistinguishability, it is possible to construct a mechanism that minimizes the quality loss (which corresponds to the inverse of the utility), using linear programming techniques. In addition we show that, under certain conditions, such mechanism also provides optimal privacy in terms of the privacy definition presented in Chapter 2 (known as the expected error of the adversary [Shokri 2012]). Furthermore, we propose a method (based on approximating distances by using spanning graphs) to reduce the number of constraints of the linear program from cubic to quadratic, maintaining the privacy guarantees and without affecting significantly the utility of the generated mechanism. This reduces considerably the time required to solve the linear program, thus enlarging significantly the location sets for which the optimal mechanisms can be computed. We end this chapter by performing a comparison with other methods in terms of privacy, utility, and performance, using data from real users to generate the probability distributions and evaluating the results.

Finally, Chapter 7 present the concluding remarks of this work.

1.3 Publications from this dissertation

The content of this dissertation is based on the following publications:

- Chapter 4 is based on the results presented in the paper **Broadening the Scope of Differential Privacy Using Metrics** [Chatzikokolakis 2013a], that appeared in the proceedings of the 13^{th} *International Privacy Enhancing Technologies Symposium (PETS 2013).*

- Chapter 5 is based on the results presented in the paper **Geo-Indisinguishability: Differential Privacy for Location-Based Systems** [Andrés 2013], that appeared in the proceedings of the 20^{th} *ACM SIGSAC Conference on Computer and Communications Security (CCS 2013).*

- Chapter 6 is based on the results presented in the paper **Optimal Geo-Indistinguishable Mechanisms for Location Privacy** [Bordenabe 2014], that appeared in the proceedings of the 21^{th} *ACM SIGSAC Conference on Computer and Communications Security (CCS 2014).*

Chapter 2

Preliminaries

In this chapter we introduce some definitions and known results from the literature that will be used throughout the rest of this thesis.

2.1 Probability measures, mechanisms and metrics

A *σ-algebra* over a set $\mathcal{X}$ is a collection $\mathcal{F}$ of subsets of $\mathcal{X}$ closed under complement and countable union, and such that $\mathcal{X} \in \mathcal{F}$. A *measure* over $(\mathcal{X}, \mathcal{F})$ is a function $\nu : \mathcal{F} \mapsto [0, \infty]$ that is countably additive and such that $\nu(\emptyset) = 0$. Common examples are the *Lebesgue measure* on $\mathbb{R}^n$, corresponding to the notions of area and volume, and the *counting measure* on countable sets, defined as $\nu(X) = |X|$. For a function $f : \mathcal{X} \to \mathbb{R}$, both $\int_X f d\nu$ and $\int_X f(x) d\nu(x)$ denote the (Lebesgue) integral of f over $X \in \mathcal{F}$ w.r.t. ν. Note that $\int_X f d\nu$ corresponds (under conditions) to the standard Riemann integral when ν is the Lebesque measure, while $\int_X f d\nu = \sum_{x \in X} f(x)$ when ν is the counting measure.

A *probability measure* is a measure μ over $(\Omega, \mathcal{F})$ (where Ω is called the sample space), such that $\mu(\Omega) = 1$. A probability measure is called *discrete* if Ω is countable and $\mathcal{F} = 2^\Omega$; in such case it can be uniquely described by the probability $\mu(\{\omega\})$ that it assigns to singleton elements $\omega \in \Omega$. In this thesis we generally assume that each sample set Ω is equipped with some natural σ-algebra $\mathcal{F}_\Omega$, which should be clear from the context. For example this would be the powerset if Ω is countable, the usual Borel algebra if $\Omega = \mathbb{R}^n$, etc. We denote by $\mathcal{P}(\Omega)$ the set of probability measures over $(\Omega, \mathcal{F}_\Omega)$. For $A, B \in \mathcal{F}$ with $\mu(B) > 0$ we define conditional probability as $\mu(A|B) = \mu(A \cap B)/\mu(B)$.

A common way of defining a probability measure μ on $(\Omega, \mathcal{F})$ is by means of a *probability density function* (pdf), that is a function $f : \Omega \to [0, \infty)$ such that $\int_\Omega f d\nu = 1$ for some reference measure ν on $(\Omega, \mathcal{F})$. In this case μ is defined as $\mu(X) = \int_X f d\nu, X \in \mathcal{F}$. We denote by $\mathcal{D}(\Omega)$ the set of pdfs over Ω.

Given two sets $\mathcal{X}$ and $\mathcal{Z}$, let $\mathcal{F}_\mathcal{Z}$ be a σ-algebra over $\mathcal{Z}$ and let $\mathcal{P}(\mathcal{Z})$ be the set of probability measures over $\mathcal{Z}$. A *mechanism* from $\mathcal{X}$ to $\mathcal{Z}$ is a (probabilistic) function $K : \mathcal{X} \to \mathcal{P}(\mathcal{Z})$. The composition $H \circ f$ of a deterministic function $f : \mathcal{X} \to \mathcal{Y}$ (called a *query*) and a mechanism $H : \mathcal{Y} \to \mathcal{P}(\mathcal{Z})$ is the mechanism $K : \mathcal{X} \to \mathcal{P}(\mathcal{Z})$ defined as $K(x) = (H \circ f)(x) = H(f(x))$. Mechanisms of this form are called *oblivious*.

Let π be a discrete probability measure on $\mathcal{X}$, called a *prior*.[1] Starting from π and using Bayes' rule, each observation $Z \in \mathcal{Z}$ of a mechanism $K : \mathcal{X} \to \mathcal{P}(\mathcal{Z})$

[1] We restrict to discrete priors for simplicity; all results could be carried to the continuous case.

induces a *posterior* measure $\sigma = \mathbf{Bayes}(\pi, K, Z)$ on $\mathcal{X}$, defined as

$$\sigma(x) = \frac{K(x)(Z)\pi(x)}{\sum_{x' \in \mathcal{X}} K(x')(Z)\pi(x')}$$

A metric on a set $\mathcal{X}$ is a function $d_{\mathcal{X}} : \mathcal{X}^2 \to [0, \infty]$ such that $d_{\mathcal{X}}(x, y) = 0$ iff $x = y$, $d_{\mathcal{X}}(x, y) = d_{\mathcal{X}}(y, x)$, and $d_{\mathcal{X}}(x, z) \leq d_{\mathcal{X}}(x, y) + d_{\mathcal{X}}(y, x)$ for all $x, y, z \in \mathcal{X}$. The *diameter* of $A \subseteq \mathcal{X}$ is defined as $d_{\mathcal{X}}(A) = \sup_{x,x' \in A} d_{\mathcal{X}}(x, x')$.

A sequence $x_1, \ldots, x_n$ is called a *chain* from x_1 to x_n and denoted by $\tilde{x}$. The length $d_{\mathcal{X}}(\tilde{x})$ of a chain is defined as $d_{\mathcal{X}}(\tilde{x}) = \sum_{i=1}^{n-1} d_{\mathcal{X}}(x_i, x_{i+1})$. If $d_{\mathcal{X}}(\tilde{x}) = d_{\mathcal{X}}(x_1, x_n)$ then $\tilde{x}$ is called *tight*.

Of particular interest are metrics *induced by a graph* $(\mathcal{X}, \sim_{\mathcal{X}})$, where $\sim_{\mathcal{X}}$ is the graph's adjacency relation. In the induced metric, $d_{\mathcal{X}}(x, x')$ is the length of the shortest path from x to x' (or infinite if no path exists). Of great interest are also the Manhattan (or L_1), the Euclidean (or L_2) and the Maximum (or L_∞) metrics, denoted by d_1, d_2, d_∞ respectively. The numerical distance on the reals (which coincides with all d_1, d_2, d_∞) will be denoted by $d_{\mathbb{R}}$ for clarity. Finally, of great interest is the metric $d_{\mathcal{P}}$ on $\mathcal{P}(\mathcal{Z})$ defined as

$$d_{\mathcal{P}}(\mu_1, \mu_2) = \sup_{Z \in \mathcal{F}_{\mathcal{Z}}} \left| \ln \frac{\mu_1(Z)}{\mu_2(Z)} \right|$$

with the convention that $\left| \ln \frac{\mu_1(Z)}{\mu_2(Z)} \right| = 0$ if both $\mu_1(Z), \mu_2(Z)$ are zero and ∞ if only one of them is zero.

2.2 Differential Privacy

Differential privacy is typically defined on databases and requires that changes to a single individual in the database should have minor effect on the outcome of a query[2]. We fix a finite domain of values $\mathcal{V}$, called the *universe*. A database $x \in \mathcal{V}^n$ consists of n records from $\mathcal{V}$ - each corresponding to an individual - that is x is a tuple $\langle x[1], \ldots, x[n] \rangle, x[i] \in \mathcal{V}$, where $x[i]$ is the value of the i-th individual in the database. We denote by $x[^v/_i]$ the database obtained from x by substituting the value v for individual i. The case when individuals are allowed to be absent from the database can be modeled by the universe $\mathcal{V}_\varnothing = \mathcal{V} \cup \{\varnothing\}$ where the null value $\varnothing$ denotes absence.

A crucial notion for differential privacy is that of *adjacency*: two databases x, x' are adjacent, written $x \sim_h x'$, if they differ in exactly one element. Let d_h be the distance induced by $\sim_h$ (i.e., $d_h(x, x')$ is the number of elements in which x, x' differ). The graph $(\mathcal{V}^n, \sim_h)$ is known as *Hamming graph*, and d_h as Hamming distance.

Let $\mathcal{Z}$ be a set of query outcomes; a mechanism $K : \mathcal{V}^n \to \mathcal{P}(\mathcal{Z})$ satisfies ϵ-differential privacy if adjacent databases produce answers with probabilities that differ

[2]An alternative definition requires the inclusion or exclusion of a single individual to have a minor efect on the query outcome. We will see in Chapter 4 that these two definitions coincide.

at most by a factor e^ϵ:

$$K(x)(Z) \leq e^\epsilon \, K(x')(Z) \quad \forall x \sim_h x' \in \mathcal{V}^n, Z \in \mathcal{F}_\mathcal{Z} \tag{2.1}$$

Following [Reed 2010], the definition can be expressed as $d_\mathcal{P}(K(x), K(x')) \leq \epsilon$ for all $x \sim_h x'$. Moreover, we can rewrite it in terms of the Hamming distance: $d_\mathcal{P}(K(x), K(x')) \leq \epsilon d_h(x, x')$ for all $x, x' \in \mathcal{V}^n$.

The following simple lemma states that bounding $d_\mathcal{P}$ is equivalent to the usual formulation of bounding the ratio between probabilities.

Lemma 2.1. *Let μ_1, μ_2 be probability measures on $\mathcal{Z}$. Then*

$$d_\mathcal{P}(\mu_1, \mu_2) \leq b \quad \textit{iff} \quad \forall Z \in \mathcal{F}_\mathcal{Z} : e^{-b}\mu_2(Z) \leq \mu_1(Z) \leq e^b \mu_2(Z)$$

Proof. ($\Rightarrow$) We have $|\ln \frac{\mu_1(Z)}{\mu_2(Z)}| \leq d_\mathcal{P}(\mu_1, \mu_2) \leq b$, hence $-b \leq \ln \frac{\mu_1(Z)}{\mu_2(Z)} \leq b$, which implies $e^{-b} \leq \frac{\mu_1(Z)}{\mu_2(Z)} \leq e^b$. ($\Leftarrow$) We have that $|\ln \frac{\mu_1(Z)}{\mu_2(Z)}|$ is bounded from above by b, but $d_\mathcal{P}(\mu_1, \mu_2)$ is the least of such bounds hence $d_\mathcal{P}(\mu_1, \mu_2) \leq b$. □

A desirable feature of this definition is that it solely depends on the mechanism itself, without explicitly talking about the adversary's side knowledge, or the information that he learns from the reported answer. However, in order to get a better understanding of a privacy definition, it is useful to give an "operational" (or "semantic") interpretation that directly restricts the abilities of the adversary. To this end, we capture the adversary's side knowledge by a prior distribution π on $\mathcal{V}^n$, and his conclusions after observing Z by the posterior distribution $\sigma = \mathbf{Bayes}(\pi, K, Z)$.

2.2.1 Operational characterizations

There are two operational interpretations commonly given to differential privacy. The first can be informally stated as: "regardless of side knowledge, by observing the reported answer an adversary obtains the same information whether or not the individual's data were included in the database". This can be formalized as follows: consider a *hiding* function $\phi_{i,v} : \mathcal{V}^n \to \mathcal{V}^n$ replacing i's value by a fixed value v, i.e. $\phi_{i,v}(x) = x[^v/_i]$, and let $\Phi_h = \{\phi_{i,v} \mid i \in 1..n, v \in \mathcal{V}\}$ be the set of all such functions. The mechanism $K \circ \phi_{i,v}$ behaves as K after removing i's value; hence we require the posterior distributions induced by $K, K \circ \phi_{i,v}$ to be similar. The resulting notion (called "semantic privacy" in [Ganta 2008])[3] can be shown to be implied by differential privacy.

Theorem 2.1. *If a mechanism $K : \mathcal{V}^n \to \mathcal{P}(\mathcal{Z})$ satisfies ϵ-differential privacy then for all priors π on $\mathcal{V}^n$, all $\phi \in \Phi_h$, and all $Z \in \mathcal{F}_\mathcal{Z}$:*

$$d_\mathcal{P}(\sigma_1, \sigma_2) \leq 2\epsilon \qquad \textit{where } \sigma_1 = \mathbf{Bayes}(\pi, K, Z) \textit{ and } \sigma_2 = \mathbf{Bayes}(\pi, K \circ \phi, Z)$$

[3] The only difference between the semantic privacy of [Ganta 2008] and our formulation is that an "additive" metric between distributions is used instead of the "multiplicative" $d_\mathcal{P}$.

Note that the above interpretation compares two *posterior* measures. This requirement does not imply that the adversary learns no information, but that he learns the same regardless of the presence of the individual's data. Both σ_1, σ_2 can be very different than the prior π, as the well-known example of Terry Gross [Dwork 2006a] demonstrates.

A different interpretation can be obtained by comparing the posterior σ to the prior distribution π. Of course, we cannot expect those to be similar, since some information is allowed to be disclosed. Still, we can require the distributions to be similar when restricted to the value of a single individual, by assuming an informed adversary who knows all other values in the database. Let $N_i(x) = \{x[^v/_i] \mid v \in \mathcal{V}\}$ denote the set of databases obtained from x by modifying i's value, and let $\mathcal{N}_h = \{N_i(x) \mid x \in \mathcal{V}^n, i \in 1..n\}$. Knowing that the database belongs to a set $N \in \mathcal{N}_h$ means that we know all values except one. We denote by $\pi_{|N}$ the distribution obtained from π by restricting to N, i.e. $\pi_{|N}(x) = \pi(x|N)$. Requiring $\pi_{|N}, \sigma_{|N}$ to be similar brings us the definition of "semantic security" from [Dwork 2006b], which is a full characterization of differential privacy.

Theorem 2.2. *A mechanism $K : \mathcal{V}^n \to \mathcal{P}(\mathcal{Z})$ satisfies ϵ-differential privacy iff for all priors π on $\mathcal{V}^n$, all $N \in \mathcal{N}_h$, and all $Z \in \mathcal{F}_\mathcal{Z}$:*

$$d_\mathcal{P}(\pi_{|N}, \sigma_{|N}) \leq \epsilon \qquad \textit{where} \qquad \sigma = \mathbf{Bayes}(\pi, K, Z)$$

Note that if the adversary does not know $N \in \mathcal{N}_h$, then his knowledge can (and will in most cases) be increased. Note also that the above result does not imply that K allows the adversary to learn $N_i(x)$. In fact, this is clearly forbidden since it would violate the same condition for $N_j(x), j \neq i$, i.e. it would violate the other individuals' privacy.

2.2.2 The Laplace mechanism

We consider a query to be a function $f : \mathcal{V}^n \to \mathcal{Y}$, with $\mathcal{Y}$ being the set of possible results with a corresponding metric $d_\mathcal{Y}$. The *sensitivity* of f is the maximum difference in the result of the query that can be obtained by modifying a single value in a database. Namely:

Definition 2.1. *A query f is Δ-sensitive iff:*

$$d_\mathcal{Y}(f(x), f(x')) \leq \Delta \quad \forall x \sim_h x' \in \mathcal{V}^n$$

The smallest such Δ (if exists) is called the sensitivity *of f.*

In order to answer queries with differential privacy guarantees, we can compose f with a mechanism $H : \mathcal{Y} \to \mathcal{P}(\mathcal{Z})$. This way, we obtain an oblivious mechanism $H \circ f : \mathcal{V}^n \to \mathcal{P}(\mathcal{Z})$,

Fact 2.1. *If f is Δ-sensitive and H satisfies ϵ-differential privacy, then $H \circ f$ satisfies $\Delta\epsilon$-differential privacy.*

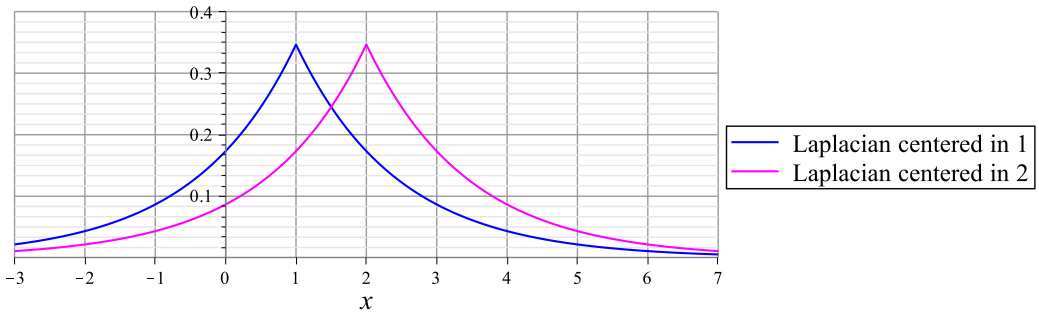


Figure 2.1: Linear laplacians with different means and same scale, representing the distributions of the reported value when the answer of the query is either 1 or 2.

For the most common case when $\mathcal{Y} = \mathcal{Z} = \mathbb{R}$, the usual technique to achieve differential privacy is by adding noise generated from a Laplace distribution with mean 0 and scale $\frac{\Delta}{\epsilon}$, which has the following pdf:

$$f(z) = \frac{\epsilon}{2\Delta} e^{-\frac{\epsilon|z|}{\Delta}}$$

Figure 2.1 shows the pdf corresponding to two different query results.

Theorem 2.3. *Let $f : \mathcal{V}^n \to \mathcal{Y}$ be a Δ-sensitive query and $K : \mathcal{Y} \to \mathcal{P}(\mathcal{Z})$ a mechanism adding noise generated from a Laplace distribution with mean 0 and scale Δ/ϵ. Then $K \circ f$ satisfies ϵ-differential privacy.*

Chapter 3

Metrics for Location Privacy

The use of location-based services (LBSs) has been significantly increased by the growing popularity of mobile devices like smartphones and tablets, in combination with the increasing availability of wireless data connections. However, while these systems have demonstrated to provide enormous benefits to individuals and society, the growing exposure of users' location information raises important privacy issues. Consider a user located in Paris who wishes to query an LBS provider for nearby restaurants. In order to keep his location information private, but at the same time obtain useful results, he is willing to disclose some approximate information z instead of his exact location x. However, it is clear that if he expects to have a reasonable level of privacy, this approximate location cannot be generated naively. Our goal is to provide a *formal* notion of privacy that adequately captures the user's expected privacy.

In this chapter, we briefly examine various notions and mechanisms from the literature in location privacy of LBSs. We pay particular attention to the notion of *expected distance* with respect to an obfuscation mechanism, which serves as the basis for two important notions used later in this thesis: the expected error of the adversary, used to quantify the location privacy provided by a mechanism, and the quality loss, used to calculate its utility.

We also explore other location privacy notions, addressing their strengths and weaknesses. This will help in defining the goals our desired privacy notion needs to satisfy.

3.1 Location obfuscation, quality loss and adversary's error

A common way of achieving location privacy is to apply a *location obfuscation* mechanism, that is a probabilistic function $K : \mathcal{X} \to \mathcal{P}(\mathcal{X})$ where $\mathcal{X}$ is the set of possible locations, and $\mathcal{P}(\mathcal{X})$ denotes the set of probability distributions over $\mathcal{X}$. K takes a location x as input, and produces a *reported location* z which is communicated to the service provider. In this paper we generally consider $\mathcal{X}$ to be finite, in which case K can be represented by a stochastic matrix, where k_{xz} is the probability to report z from location x.

A prior distribution $\pi \in \mathcal{P}(\mathcal{X})$ on the set of locations can be viewed either as modelling the behaviour of the user (the *user profile*), or as capturing the adversary's *side information* about the user. Given a prior π and a metric d on $\mathcal{X}$, the expected

distance between the real and the reported location is:

$$\textsc{ExpDist}(K, \pi, d) = \sum_{x,z} \pi_x k_{xz} d(x, z)$$

From the user's point of view, we want to quantify the service *quality loss (QL)* produced by the mechanism K. Given a *quality metric* d_Q on locations, such that $d_Q(x, z)$ measures how much the quality decreases by reporting z when the real location is x (the Euclidean metric d_2 being a typical choice), we can naturally define the quality loss as the expected distance between the real and the reported location, that is

$$\mathrm{QL}(K, \pi, d_Q) = \textsc{ExpDist}(K, \pi, d_Q)$$

The QL can also be viewed as the (inverse of the) utility of the mechanism.

Similarly, we want to quantify the *privacy* provided by K. A natural approach is to consider a Bayesian adversary with some prior information π, trying to remap z back to a guessed location $\hat{x}$. A remapping strategy can be modelled by a stochastic matrix H, where $h_{z\hat{x}}$ is the probability to map z to $\hat{x}$. Then the privacy of the mechanism can be defined as the expected error of an adversary under the best possible remapping [Shokri 2011, Shokri 2012, Hoh 2005]:

$$\textsc{AdvError}(K, \pi, d_A) = \min_H \textsc{ExpDist}(KH, \pi, d_A)$$

Note that the composition KH of K and H is itself a mechanism. Similarly to d_Q, the metric $d_A(x, \hat{x})$ captures the adversary's loss when he guesses $\hat{x}$ while the real location is x. Note that d_Q and d_A can be different, but the canonical choice is to use the Euclidean distance for both. However, in this thesis we do not make any assumption about what these metrics are.

A natural question, then, is to construct a mechanism that achieves *optimal privacy*, given a *QL constraint*.

Definition 3.1. *Given a prior π, a quality metric d_Q, a quality bound q and an adversary metric d_A, a mechanism K is q-*$\textsc{OptPriv}(\pi, d_A, d_Q)$ *iff*

1. $\mathrm{QL}(K, \pi, d_Q) \leq q$, *and*
2. *for all mechanisms K',* $\mathrm{QL}(K', \pi, d_Q) \leq q$ *implies* $\textsc{AdvError}(K', \pi, d_A) \leq \textsc{AdvError}(K, \pi, d_A)$

In other words, a q-$\textsc{OptPriv}$ mechanism provides the best privacy (expressed in terms of $\textsc{AdvError}$) among all mechanisms with QL at most q. This problem was studied in [Shokri 2012], providing a method to construct such a mechanism for any q, π, d_A, d_Q, by solving a zero-sum Bayesian Stackelberg game with a properly constructed linear program.

It is worth noting that this privacy notion and the obfuscation mechanisms based on it are explicitly defined in terms of the adversary's side information. This implies that location-obfuscation mechanisms based on this notion assume that the attacker have some particular kind of side-information (for instance, past location traces of the user), and therefore the definition is only satisfied for this limited class of adversaries.

3.2 Other ways to measure location privacy

k-anonymity

The notion of k-anonymity is the most widely used definition of privacy for location-based systems in the literature. Many systems in this category [Gruteser 2003, Gedik 2005, Mokbel 2006] aim at protecting the user's *identity*, requiring that the attacker cannot infer which user is executing the query, among a set of k different users. Such systems are outside the scope of our problem, since we are interested in protecting the user's *location*.

On the other hand, k-anonymity has also been used to protect the user's location (sometimes called l-diversity in this context), requiring that it is indistinguishable among a set of k points (often required to share some semantic property). One way to achieve this is through the use of *dummy locations* [Kido 2005, Shankar 2009]. This technique involves generating $k-1$ properly selected dummy points, and performing k queries to the service provider, using the real and dummy locations. Another method for achieving k-anonymity is through *cloaking* [Bamba 2008, Duckham 2005, Xue 2009]. This involves creating a cloaking region that includes k points sharing some property of interest, and then querying the service provider for this cloaking region.

Even when side knowledge does not explicitly appear in the definition of k-anonymity, a system cannot be proven to satisfy this notion unless assumptions are made about the attacker's side information. For example, dummy locations are only useful if they look equally likely to be the real location from the point of view of the attacker. Any side information that allows to rule out any of those points, as having low probability of being the real location, would immediately violate the definition.

Counter-measures are often employed to avoid this issue: for instance, [Kido 2005] takes into account concepts such as ubiquity, congestion and uniformity for generating dummy points, in an effort to make them look realistic. Similarly, [Xue 2009] takes into account the user's side information to construct a cloaking region. Such counter-measures have their own drawbacks: first, they complicate the employed techniques, also requiring additional data to be taken into account (for instance, precise information about the environment or the location of nearby users), making their application in real-time by a handheld device challenging. Moreover, the attacker's actual side information might simply be inconsistent with the assumptions being made. A detailed study of the flaws of k-anonymity as a framework for location privacy have also been studied in [Shokri 2010].

As a result, notions that abstract from the attacker's side information, such as differential privacy, have been growing in popularity in recent years, compared to k-anonymity-based approaches.

Differential Privacy

Differential privacy has also been used in the context of location privacy. In the work of [Machanavajjhala 2008], it is shown that a synthetic data generation tech-

nique can be used to publish statistical information about commuting patterns in a differentially private way. In [Ho 2011], a quadtree spatial decomposition technique is used to ensure differential privacy in a database with location pattern mining capabilities.

As shown in the aforementioned works, differential privacy can be successfully applied in cases where *aggregate* information about several users is published. On the other hand, the nature of this notion makes it poorly suitable for applications in which only a single individual is involved, such as our motivating scenario. The secret in this case is the location of a single user. Thus, differential privacy would require that any change in that location should have negligible effect on the published output, making it impossible to communicate any useful information to the service provider.

To overcome this issue, Dewri [Dewri 2012] proposes a mix of differential privacy and k-anonymity, by fixing an anonymity set of k locations and requiring that the probability to report the same obfuscated location z from any of these k locations should be similar (up to e^{ϵ}). This property is achieved by adding Laplace noise to each Cartesian coordinate independently. There are however two problems with this definition: first, the choice of the anonymity set crucially affects the resulting privacy; outside this set no privacy is guaranteed at all. Second, the property itself is rather weak; reporting the geometric median (or any deterministic function) of the k locations would satisfy the same definition, although the privacy guarantee would be substantially lower than using Laplace noise.

Nevertheless, Dewri's intuition of using Laplace noise[1] for location privacy is valid, and [Dewri 2012] provides extensive experimental analysis supporting this claim.

Approach-specific location-privacy metrics

There are also other location-privacy definitions that can be found in the literature, usually specific to some particular obfuscation mechanism. [Cheng 2006] proposes a location cloaking mechanism, and focuses on the evaluation of Location-based Range Queries. The degree of privacy is measured by the size of the cloak (also called *uncertainty region*), and by the coverage of sensitive regions, which is the ratio between the area of the cloak and the area of the regions inside the cloak that the user considers to be sensitive. In order to deal with the side-information that the attacker may have, ad-hoc solutions are proposed, like patching cloaks to enlarge the uncertainty region or delaying requests. Both solutions may cause a degradation in the quality of service.

In [Ardagna 2007], the real location of the user is assumed to have some level of inaccuracy, due to the specific sensing technology or to the environmental conditions. Different obfuscation techniques are then used to increase this inaccuracy in order

[1] The planar Laplace distribution that we use later in this thesis, however, is different from the distribution obtained by adding Laplace noise to each Cartesian coordinate, and has better differential privacy properties (c.f. Section 5.2).

to achieve a certain level of privacy. This level of privacy is computed as (the opposite of) the *relevance* of the location measurement. Relevance is defined as the ratio between the accuracy before and after the application of the obfuscation techniques.

Similar to the case of k-anonymity, both privacy metrics mentioned above make implicit assumptions about the adversary's side information. This may imply a violation of the privacy definition in a scenario where the adversary has some knowledge (maybe probabilistic) about the user's real location.

Transformation-based approaches

A number of approaches for location privacy are radically different from the ones mentioned so far. Instead of cloaking the user's location, they aim at making it completely invisible to the service provider. This is achieved by transforming all data to a different space, usually employing cryptographic techniques, so that they can be mapped back to spatial information only by the user [Khoshgozaran 2007, Ghinita 2008]. The data stored in the provider, as well as the location send by the user are encrypted. Then, using techniques from *private information retrieval*, the provider can return information about the encrypted location, without ever discovering which actual location it corresponds to.

A drawback of these techniques is that they are computationally demanding, making it difficult to implement them in a handheld device. Moreover, they require the provider's data to be encrypted, making it impossible to use existing providers, such as Google Maps, which have access to the real data.

Chapter 4

Generalizing Privacy with Metrics

Because of the focus on the single individual as the unit of protection, differential privacy relies in a crucial way on the notion of two databases being *adjacent*, i.e. differing only for an individual. For two non-adjacent databases, there is no requirement other than the one induced by the transitive application of the property. When the sensitive information to be protected is other than the value of a single individual, it is common to consider different notions of adjacency. For example, in cases of cohesive groups with highly correlated values, we could consider adjacent two databases differing in any number of individuals of the same group. Similarly, when dealing with friendship graphs in social networks, adjacency could be defined as differing in a single edge.

We argue that in some situations the distinguishability level between x and x' should depend not only on the number of different values between x and x', but also on the values themselves. We might require, for instance, databases in which the value of an individual is only slightly modified to be highly indistinguishable, thus protecting the *accuracy* by which an analyst can infer an individual's value.

More generally, we might want to apply differential privacy in scenarios when x, x' are not databases at all, but belong to an *arbitrary domain of secrets* $\mathcal{X}$. In such a scenario, there might be no natural notion of adjacency, but it is still reasonable to define a distinguishability level between secrets, and employ the same principle of differential privacy – i.e. the smaller the distinguishability level between x, x' is, the more similar the probability distributions $K(x)$, $K(x')$ are required to be – to obtain a meaningful notion of privacy.

In the case of an arbitrary set of secrets $\mathcal{X}$, equipped with a metric $d_{\mathcal{X}}$, differential privacy can be generalized as follows:

Definition 4.1. *A mechanism $K : \mathcal{X} \to \mathcal{P}(\mathcal{Z})$ satisfies $d_{\mathcal{X}}$-privacy, iff $\forall x, x' \in \mathcal{X}$: $d_{\mathcal{P}}(K(x), K(x')) \leq d_{\mathcal{X}}(x, x')$, or equivalently:*

$$K(x)(Z) \leq e^{d_{\mathcal{X}}(x,x')}\, K(x')(Z) \quad \forall Z \in \mathcal{F}_{\mathcal{Z}}$$

Intuitively, the definition requires that secrets close to each other w.r.t. $d_{\mathcal{X}}$, meaning hardly distinguishable, should produce outcomes with similar probability. This is the same core idea as in differential privacy, which can be retrieved as $\mathcal{X} = \mathcal{V}^n, d_{\mathcal{X}} = \epsilon d_h$.

Note that Definition 4.1 contains no ϵ; the distinguishability level is directly given by the metric. In practice, the desired metric can be obtained from a standard one by scaling by a proper factor ϵ (recall that a scaled metric is also a metric). For

instance, in the case of standard differential privacy, the Hamming distance between adjacent databases is 1, and we want their distinguishability level to be ϵ, hence we use the scaled version ϵd_h.

Note also that an *extended* metric (allowing $d_{\mathcal{X}}(x, x') = \infty$) can be useful in cases when we allow two secrets to be completely distinguished. The understanding of Definition 4.1 is that the requirement is always satisfied for those secrets. Similarly, *pseudo*-metrics (allowing $d_{\mathcal{X}}(x, x') = 0$ for $x \neq x'$) could be useful when we want some secrets to be completely indistinguishable (forcing $K(x)$ and $K(x')$ to be identical). To simplify the presentation, in this thesis assume an extended metric (but not pseudo).

Different metrics $d_{\mathcal{X}}, d_{\mathcal{X}}{}'$ on the same set $\mathcal{X}$ clearly give rise to different privacy notions. The "strength" of each notion depends on the distinguishability level assigned to each pair of secrets; $d_{\mathcal{X}}$-privacy and $d_{\mathcal{X}}{}'$-privacy are in general incomparable. However, lower distinguishability level implies stronger privacy.

Proposition 4.1. *If $d_{\mathcal{X}} \leq d_{\mathcal{X}}{}'$ (point-wise) then $d_{\mathcal{X}}$-privacy implies $d_{\mathcal{X}}{}'$-privacy.*

Proof. Immediate, since $d_{\mathcal{P}}(K(x), K(x')) \leq d_{\mathcal{X}}(x, x') \leq d_{\mathcal{X}}{}'(x, x')$. □

For example, some works consider an adjacency relation $\sim_r$ slightly different than $\sim_h$, defined as $x \sim_r x'$ iff $x' = x[^{\varnothing}/_i]$ (or vice versa), i.e. x' can be obtained from x by removing one individual. This relation gives rise to a metric d_r for which it holds that: $\frac{1}{2} d_r \leq d_h \leq d_r$. From Proposition 4.1, the two models are essentially equivalent; one can obtain ϵd_r-privacy from ϵd_h-privacy by doubling ϵ and vice versa.

4.1 Operational characterizations

Similarly to standard differential privacy, $d_{\mathcal{X}}$-privacy does not explicitly talk about the adversary's gain of knowledge. To better understand the privacy guarantees provided by a certain metric $d_{\mathcal{X}}$, it is useful to directly reason about the capabilities of the adversary. Two such characterizations are given, generalizing the two interpretations of standard differential privacy (Theorems 2.1,2.2).

4.1.1 First characterization

The first characterization uses the concept of a *hiding* function $\phi : \mathcal{X} \to \mathcal{X}$. The idea is that ϕ can be applied to x before the mechanism K, so that the latter has only access to a hidden version $\phi(x)$, instead of the real secret x. Let $d_{\mathcal{X}}(\phi) = \sup_{x \in \mathcal{X}} d_{\mathcal{X}}(x, \phi(x))$ be the maximum distance between a secret and its hidden version. We can show that $d_{\mathcal{X}}$-privacy implies that the adversary's conclusions (captured by his posterior measure) are the same (up to $2d_{\mathcal{X}}(\phi)$) regardless of whether ϕ is applied or not. Moreover, we show that certain classes of hiding functions are "canonical", in the sense that if the property holds for those, it must hold in general. We start by defining this class.

Definition 4.2. *Let Φ be a set of functions from $\mathcal{X}$ to $\mathcal{X}$, called* hiding *functions. A chain $\tilde{x}$ is called a* maximal Φ-chain *iff for every step i there exists $\phi \in \Phi$ s.t. $\phi(x_i) = x_{i+1}, \phi(x_{i+1}) = x_i$ and $d_{\mathcal{X}}(x_i, x_{i+1}) = d_{\mathcal{X}}(\phi)$. Then Φ is called* maximally tight *w.r.t. $d_{\mathcal{X}}$ iff $\forall x, x' \in \mathcal{X}$ there exists a tight maximal Φ-chain from x to x'.*

Note that the above property requires hiding functions that *swap* the secrets x_i, x_{i+1}. This is not satisfied by the hiding functions $\phi_{i,v}$ introduced in Section 2.2.1, but will be satisfied by more general functions used later in this thesis.

The following Lemma shows the usefulness of *tight* chains.

Lemma 4.1. *Let $x_1, \ldots, x_n$ be a tight chain. If K satisfies $d_{\mathcal{X}}$-privacy on all adjacent elements of the chain, then it also satisfies it for x_1, x_n. That is*

$$d_{\mathcal{P}}(K(x_i), K(x_{i+1})) \leq d_{\mathcal{X}}(x_i, x_{i+1}) \quad \forall 1 \leq i < n$$

implies $d_{\mathcal{P}}(K(x_1), K(x_n)) \leq d_{\mathcal{X}}(x_1, x_n)$.

Proof. Using the fact that $d_{\mathcal{P}}$ is itself a metric, we have

$$\begin{aligned} d_{\mathcal{P}}(K(x_1), K(x_n)) &\leq \textstyle\sum_{i=1}^{n-1} d_{\mathcal{P}}(K(x_i), K(x_{i+1})) && \text{triangle ineq. for } d_{\mathcal{P}} \\ &\leq \textstyle\sum_{i=1}^{n-1} d_{\mathcal{X}}(x_i, x_{i+1}) && \text{hypothesis} \\ &= d_{\mathcal{X}}(x_1, x_n) && \text{tightness} \end{aligned}$$

□

Theorem 4.1. *Let Φ be a set of hiding functions. If K satisfies $d_{\mathcal{X}}$-privacy then for all $\phi \in \Phi$, all priors π on $\mathcal{X}$, and all $Z \in \mathcal{F}_{\mathcal{Z}}$:*

$$d_{\mathcal{P}}(\sigma_1, \sigma_2) \leq 2\, d_{\mathcal{X}}(\phi) \qquad \text{where} \qquad \begin{aligned} \sigma_1 &= \mathbf{Bayes}(\pi, K, Z) \\ \sigma_2 &= \mathbf{Bayes}(\pi, K \circ \phi, Z) \end{aligned}$$

If Φ is maximally tight then the converse also holds.

Proof. Assume that K satisfies $d_{\mathcal{X}}$-privacy and let π be a prior, $\phi \in \Phi$ and $Z \in \mathcal{F}_{\mathcal{Z}}$. We need to show that

$$\forall x \in \mathcal{X} : e^{-2d_{\mathcal{X}}(\phi)}\sigma_1(x) \leq \sigma_2(x) \leq e^{2d_{\mathcal{X}}(\phi)}\sigma_1(x)$$

(then conclude by applying Lemma 2.1). Let $x \in \mathcal{X}$, we have:

$$\begin{aligned} &\sigma_2(x) \\ &= \frac{(K \circ \phi)(x)(Z)\pi(x)}{\sum_{x' \in \mathcal{X}} (K \circ \phi)(x')(Z)\pi(x')} && \text{def. of } \mathbf{Bayes} \\ &= \frac{K(\phi(x))(Z)\pi(x)}{\sum_{x' \in \mathcal{X}} K(\phi(x'))(Z)\pi(x')} \\ &\leq \frac{e^{d_{\mathcal{X}}(x, \phi(x))} K(x)(Z)\pi(x)}{\sum_{x' \in \mathcal{X}} e^{-d_{\mathcal{X}}(x', \phi(x'))} K(x')(Z)\pi(x')} && d_{\mathcal{X}}\text{-privacy} \\ &\leq \frac{e^{d_{\mathcal{X}}(\phi)} K(x)\pi(x)(Z)}{e^{-d_{\mathcal{X}}(\phi)} \sum_{x' \in \mathcal{X}} K(x')(Z)\pi(x')} && d_{\mathcal{X}}(x, \phi(x)) \leq d_{\mathcal{X}}(\phi) \\ &\leq e^{2d_{\mathcal{X}}(\phi)}\sigma_1(x) && \text{def. of } \mathbf{Bayes} \end{aligned}$$

and symmetrically for $\sigma_2(x) \geq e^{-2d_{\mathcal{X}}(\phi)}\sigma_1(x)$.

For the opposite direction, assume that Φ is maximally tight (Def 4.2), that $d_{\mathcal{P}}(\sigma_1, \sigma_2) \leq 2d_{\mathcal{X}}(\phi)$ holds for all π, ϕ, Z, but $d_{\mathcal{X}}$-privacy is violated for some $x, x' \in \mathcal{X}$. From Def 4.2, there exist a tight maximal Φ-chain $\tilde{x}$ from x to x'. Then from Lemma 4.1, we get that $d_{\mathcal{X}}$-privacy is also violated for some adjacent x_i, x_{i+1} in the chain, that is:

$$K(x_i)(Z) > e^{d_{\mathcal{X}}(x_i, x_{i+1})} K(x_{i+1})(Z) \qquad \text{for some } Z \tag{4.1}$$

We fix Z to the one above. Since $\tilde{x}$ is a maximal Φ-chain, there exists $\phi \in \Phi$ such that $\phi(x_i) = x_{i+1}, \phi(x_{i+1}) = x_i$ and $d_{\mathcal{X}}(x_i, x_{i+1}) = d_{\mathcal{X}}(\phi)$. Fixing this ϕ, we define a function $f : \mathcal{P}(\mathcal{X}) \to \mathbb{R}$ as follows:

$$f(\pi) = \frac{\sum_{x' \in \mathcal{X}} K(x')(Z)\pi(x')}{\sum_{x' \in \mathcal{X}} K(\phi(x'))(Z)\pi(x')}$$

Let $\delta(x)$ denote the Dirac measure assigning probability 1 to x, from (4.1) we have that

$$f(\delta(x_i)) = \frac{K(x_i)(Z)}{K(x_{i+1})(Z)} > e^{-d_{\mathcal{X}}(x_i, x_{i+1})}$$
$$f(\delta(x_{i+1})) = \frac{K(x_{i+1})(Z)}{K(x_i)(Z)} < e^{-d_{\mathcal{X}}(x_i, x_{i+1})}$$

From the continuity of f on the line between $\delta(x_i)$ and $\delta(x_{i+1})$, there exists a prior $\pi = t\delta(x_i) + (1-t)\delta(x_{i+1}), t \in (0,1)$, such that $f(\pi) = e^{-d_{\mathcal{X}}(x_i, x_{i+1})}$. Note that since π is distinct from $\delta(x_i), \delta(x_{i+1})$, it holds that $\pi(x_i) > 0, \pi(x_{i+1}) > 0$. By applying the hypothesis for this π, we get

$$\begin{aligned}
d_{\mathcal{P}}(\sigma_1, \sigma_2) &\leq 2d_{\mathcal{X}}(\phi) && \Rightarrow \\
\sigma_1(x_i) &\leq e^{2d_{\mathcal{X}}(\phi)}\sigma_2(x_i) && \text{(Lemma 2.1)} \Rightarrow \\
\frac{K(x_i)(Z)\pi(x_i)}{\sum_{x' \in \mathcal{X}} K(x')(Z)\pi(x')} &\leq e^{2d_{\mathcal{X}}(\phi)} \frac{K(\phi(x_i))(Z)\pi(x_i)}{\sum_{x' \in \mathcal{X}} K(\phi(x'))(Z)\pi(x')} && \text{(Def. of } \sigma_1, \sigma_2) \Rightarrow \\
K(x_i)(Z) &\leq e^{2d_{\mathcal{X}}(\phi)} f(\pi) K(\phi(x_i))(Z) && (\pi(x_i) > 0) \Rightarrow \\
K(x_i)(Z) &\leq e^{d_{\mathcal{X}}(x_i, x_{i+1})} K(x_{i+1})(Z)
\end{aligned}$$

which contradicts (4.1). □

The above characterization compares two posterior distributions; hence, it does not impose that the adversary gains no information, but that this information is the almost the same regardless of whether ϕ has been applied to the secret or not.

4.1.2 Second characterization

A different approach is to compare the adversary's prior and posterior distributions, measuring how much he learned about the secret. Since we allow some information

to be revealed, we cannot expect these distributions to be similar. Still, if we restrict to a neighborhood N of secrets that are close to each other, we can show that $d_{\mathcal{X}}$-privacy implies that an informed adversary, knowing that the secret belongs to N, can gain little more information about the exact secret regardless of his side knowledge about N. Moreover, similarly to the previous characterization, we show that certain classes of neighborhoods are "canonical". We denote with $d_{\mathcal{X}}(N)$ the maximum distance between elements of N.

Definition 4.3. *Let $\mathcal{N} \subseteq 2^{\mathcal{X}}$. The elements of $\mathcal{N}$ are called* neighborhoods. *A chain $\tilde{x}$ is called a* maximal $\mathcal{N}$-chain *iff for every step i there exists $N \in \mathcal{N}$ such that $\{x_i, x_{i+1}\} \subseteq N$ and $d_{\mathcal{X}}(x_i, x_{i+1}) = d_{\mathcal{X}}(N)$. Then $\mathcal{N}$ is called* maximally tight *w.r.t. $d_{\mathcal{X}}$ iff $\forall x, x' \in \mathcal{X}$ there exists a tight maximal $\mathcal{N}$-chain from x to x'.*

The operational scenario is similar to the one of differential privacy. The $\mathcal{N}$-adversary (for some $\mathcal{N}$), selects a neighborhood N and a prior π on secrets. He wins the game if, after observing the output of the mechanism, his posterior probability, restricted to N, is increased by more than $d_{\mathcal{X}}(N)$.

The following result states that if K satisfies $d_{\mathcal{X}}$-privacy then no such adversary (for any $\mathcal{N}$) can win the game. Moreover, a maximally tight $\mathcal{N}$ represents a "canonical" adversary that is as powerful as all others. The incapability of such an adversary to win the game is sufficient to imply $d_{\mathcal{X}}$-privacy.

Theorem 4.2. *Let $\mathcal{N} \subseteq 2^{\mathcal{X}}$. If K satisfies $d_{\mathcal{X}}$-privacy then for all $N \in \mathcal{N}$, all priors π on $\mathcal{X}$, and all $Z \in \mathcal{F}_{\mathcal{Z}}$:*

$$d_{\mathcal{P}}(\pi_{|N}, \sigma_{|N}) \leq d_{\mathcal{X}}(N) \qquad \text{where} \qquad \sigma = \mathbf{Bayes}(\pi, K, Z)$$

If $\mathcal{N}$ is maximally tight then the converse also holds.

Proof. Assume that K satisfies $d_{\mathcal{X}}$-privacy. We fix some $N \in \mathcal{N}, \pi \in \mathcal{P}(\mathcal{X}), Z \in \mathcal{F}_{\mathcal{Z}}$ and let $\sigma = \mathbf{Bayes}(\pi, K, Z)$. Note that $\pi_{|N}, \sigma_{|N}$ are distributions on N. From Lemma 2.1 we need to show that

$$e^{-d_{\mathcal{X}}(N)}\pi_{|N}(x) \leq \sigma_{|N}(x) \leq e^{d_{\mathcal{X}}(N)}\pi_{|N}(x) \qquad \forall x \in N$$

Fixing some $x \in N$, we have:

$$\begin{aligned}
\sigma_{|N}(x) &= \sigma(x|N) && \text{def. of } \sigma_{|N} \\
&= \frac{\sigma(x)}{\sum_{x' \in N} \sigma(x')} \\
&= \frac{\pi(x)K(x)(Z)}{\sum_{x' \in N} \pi(x')K(x')(Z)} && \text{def. of } \mathbf{Bayes} \\
&\leq \frac{\pi(x)K(x)(Z)}{\sum_{x' \in N} \pi(x')e^{-d_{\mathcal{X}}(x,x')}K(x)(Z)} && d_{\mathcal{X}}\text{-privacy} \\
&\leq e^{d_{\mathcal{X}}(N)}\frac{\pi(x)}{\sum_{x' \in N} \pi(x')} && d_{\mathcal{X}}(x, x') \leq d_{\mathcal{X}}(N) \\
&= e^{d_{\mathcal{X}}(N)}\pi_{|N}(x)
\end{aligned}$$

and symmetrically for $\sigma_{|N}(x) \geq e^{-d_{\mathcal{X}}(N)}\pi_{|N}(x)$.

For the opposite direction, assume that $\mathcal{N}$ is maximally tight (Def 4.3) but $d_{\mathcal{X}}$-privacy is violated for some $x, x' \in \mathcal{X}$. From Def 4.3, there exist a tight $\mathcal{N}$-chain $\tilde{x}$ from x to x'. Then from Lemma 4.1, we get that $d_{\mathcal{X}}$-privacy is also violated for some adjacent x_i, x_{i+1} in the chain, that is:

$$K(x_i)(Z) > e^{d_{\mathcal{X}}(x_i,x_{i+1})}K(x_{i+1})(Z) \qquad \text{for some } Z \tag{4.2}$$

Since $\tilde{x}$ is an $\mathcal{N}$-chain, there exist $N \in \mathcal{N}$ such that $\{x_i, x_{i+1}\} \subseteq N$ and $d_{\mathcal{X}}(x_i, x_{i+1}) = d_{\mathcal{X}}(N)$. We define a prior distribution $\pi_t(x)$ as

$$\pi_t(x) = \begin{cases} t & x = x_i \\ 1-t & x = x_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

Using that prior for $t > 0$, we fix some $Z \in \mathcal{F}_{\mathcal{Z}}$ and let $\sigma_t = \mathbf{Bayes}(\pi_t, K, Z)$. We have

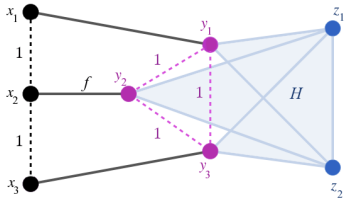
$$\begin{aligned}
\sigma_{t|N}(x_i) &\leq e^{d_{\mathcal{X}}(N)}\pi_{t|N}(x_i) && \text{(hypoth., Lemma 2.1)} \Rightarrow \\
\sigma_t(x_i) &\leq e^{d_{\mathcal{X}}(N)}\pi_t(x_i) && (\pi_t(N) = \sigma_t(N) = 1) \Rightarrow \\
\frac{\pi_t(x_i)K(x_i)(Z)}{\sum_{x' \in \mathcal{X}} \pi_t(x')K(x')(Z)} &\leq e^{d_{\mathcal{X}}(N)}\pi_t(x_i) && \text{(def. of } \mathbf{Bayes}) \Rightarrow \\
\frac{tK(x_i)(Z)}{tK(x_i)(Z) + (1-t)K(x_{i+1})(Z)} &\leq e^{d_{\mathcal{X}}(N)}t && \text{(def. of } \pi_t) \Rightarrow \\
\frac{K(x_i)(Z)}{tK(x_i)(Z) + (1-t)K(x_{i+1})(Z)} &\leq e^{d_{\mathcal{X}}(N)} && (t > 0) \Rightarrow \\
\frac{K(x_i)(Z)}{tK(x_i)(Z) + (1-t)K(x_{i+1})(Z)} &\leq e^{d_{\mathcal{X}}(x_i,x_{i+1})} && (d_{\mathcal{X}}(x_i, x_{i+1}) = d_{\mathcal{X}}(N))
\end{aligned}$$

The above inequality holds for all $t > 0$. Finally, taking the $\lim_{t\to 0}$ on both sides we get

$$K(x_i)(Z) \leq e^{d_{\mathcal{X}}(x_i,x_{i+1})}K(x_{i+1})(Z)$$

which is a contradiction of (4.2). □

Using meaningful (and maximally tight) sets $\Phi, \mathcal{N}$, and applying the above characterizations, we can get an intuitive understanding of the privacy guarantees offered by $d_{\mathcal{X}}$-privacy. For example, in the case of databases, it can be shown that $\mathcal{N}_h$ is maximally tight w.r.t. the d_h metric, hence the characterization of Theorem 2.2 can be obtained as a special case of Theorem 4.2. Theorem 2.1 can also be obtained from Theorem 4.1 (even though Φ_h is not maximally tight) since it only states an implication in one direction.



	z_1	z_2
y_1	$3/4$	$1/4$
y_2	$1/2$	$1/2$
y_3	$1/4$	$3/4$

Figure 4.1: Counterexample to the converse of Fact 4.1. The table represents the distribution H. We note that $H \circ f$ satisfies $(\ln 2)$-privacy, and that f is 1-sensitive. However $H(y_1)(z_1) = 3/4 \not\leq 2\, H(y_3)(z_1) = 1/2$, hence H does not satisfy $(\ln 2)$-privacy.

4.2 Answering Queries

To obtain the answer to a query $f : \mathcal{X} \to \mathcal{Y}$ in a private way, we can compose it with a mechanism $H : \mathcal{Y} \to \mathcal{P}(\mathcal{Z})$, thus obtaining an oblivious mechanism $H \circ f : \mathcal{X} \to \mathcal{P}(\mathcal{Z})$. In this section, we first state the standard compositionality result about the privacy of $H \circ f$, relying on the notion of Δ-*sensitivity* (aka Lipschitz continuity), naturally extended to the case of $d_{\mathcal{X}}$-privacy. Then, we introduce the concept of *uniform* sensitivity, and we use it to obtain the converse of the aforementioned compositionality result, which in turn allows to give optimality results later in the chapter.

Definition 4.4. *f is Δ-sensitive w.r.t. $d_{\mathcal{X}}, d_{\mathcal{Y}}$ iff $d_{\mathcal{Y}}(f(x), f(x')) \leq \Delta\, d_{\mathcal{X}}(x, x')$ for all $x, x' \in \mathcal{X}$. The smallest such Δ (if exists) is called the* sensitivity *of f w.r.t. $d_{\mathcal{X}}, d_{\mathcal{Y}}$.*

Fact 4.1. *Assume that f is Δ-sensitive w.r.t. $d_{\mathcal{X}}, d_{\mathcal{Y}}$ and H satisfies $d_{\mathcal{Y}}$-privacy. Then $H \circ f$ satisfies $\Delta d_{\mathcal{X}}$-privacy.*

Proof. Assume that H satisfies $d_{\mathcal{Y}}$-privacy and let $x, x' \in \mathcal{X}$. We have:

$$\begin{aligned} d_{\mathcal{P}}((H \circ f)(x), (H \circ f)(x')) &= d_{\mathcal{P}}(H(f(x)), H(f(x'))) \\ &\leq d_{\mathcal{Y}}(f(x), f(x')) && d_{\mathcal{Y}}\text{-privacy} \\ &\leq \Delta d_{\mathcal{X}}(x, x') && \Delta\text{-sensitivity} \end{aligned}$$

□

Note that it is common to define a family of mechanisms $H_\epsilon, \epsilon > 0$, instead of a single one, where each H_ϵ satisfies privacy for a scaled version $\epsilon d_{\mathcal{Y}}$ of a metric of interest $d_{\mathcal{Y}}$. Given such a family and a query f, we can define a family of oblivious mechanisms $K_\epsilon = H_{\epsilon/\Delta} \circ f, \epsilon > 0$, each satisfying $\epsilon d_{\mathcal{X}}$-privacy (from Fact 4.1).

The converse of the above result does not hold in general, see Fig. 4.1 for a counterexample. However, it does hold if we replace the notion of sensitivity by the stronger notion of *uniform sensitivity*.

Definition 4.5. *Two elements $y, y' \in \mathcal{Y}$ are called Δ-expansive iff $d_{\mathcal{Y}}(y, y') = \Delta d_{\mathcal{X}}(x, x')$ for some $x \in f^{-1}(y), x' \in f^{-1}(y')$. A chain $\tilde{y}$ is Δ-expansive iff all steps y_i, y_{i+1} are Δ-expansive. Finally, f is uniformly Δ-sensitive iff it is Δ-sensitive and for all $y, y' \in \mathcal{Y}$ there exists a tight and Δ-expansive chain from y to y'.*

Theorem 4.3. *Assume that f is uniformly Δ-sensitive w.r.t. $d_{\mathcal{X}}$, $d_{\mathcal{Y}}$. Then H satisfies $d_{\mathcal{Y}}$-privacy if and only if $H \circ f$ satisfies $\Delta d_{\mathcal{X}}$-privacy.*

Proof. The ($\Rightarrow$) part is Fact 4.1. For the ($\Leftarrow$) part, fix some $y, y' \in \mathcal{Y}$ and let $y_1, \ldots, y_n$ be the tight Δ-expansive chain from y to y' guaranteed to exist by the definition of uniform Δ-sensitivity. Then, for all $1 \leq i < n$, since y_i, y_{i+1} are Δ-expansive, there exist

$$x \in f^{-1}(y_i), x' \in f^{-1}(y_{i+1}) \quad \text{such that} \quad d_{\mathcal{Y}}(f(x), f(x')) = \Delta\, d_{\mathcal{X}}(x, x')$$

Hence

$$\begin{aligned} d_{\mathcal{P}}(H(y_i), H(y_{i+1})) &= d_{\mathcal{P}}(H(f(x)), H(f(x'))) \\ &\leq \Delta d_{\mathcal{X}}(x, x') && \Delta d_{\mathcal{X}}\text{-privacy of } H \circ f \\ &= d_{\mathcal{Y}}(y_i, y_{i+1}) \end{aligned}$$

So H satisfies $d_{\mathcal{Y}}$-privacy for all adjacent elements in the chain, hence from Lemma 4.1 it also satisfies it for y, y'. □

4.2.1 Laplace Mechanisms

Adding Laplace noise is the most widely used technique for achieving differential privacy. The mechanism can be naturally adapted to any metric, using a variant of the exponential mechanism [McSherry 2007], by providing a properly constructed scaling function. Note that in the framework of d-privacy, we can express the privacy of the mechanism itself, on its own domain, without the need to consider a query or a notion of sensitivity.

Definition 4.6. *Let $\mathcal{Y}, \mathcal{Z}$ be two sets, and let $d_{\mathcal{Y}}$ be a metric on $\mathcal{Y} \cup \mathcal{Z}$. Let $\lambda : \mathcal{Z} \to [0, \infty)$ be a scaling function such that $D(y)(z) = \lambda(z)\, e^{-d_{\mathcal{Y}}(y,z)}$ is a pdf for all $y \in \mathcal{Y}$ (i.e. $\int_{\mathcal{Z}} D(y)(z) d\nu(z) = 1$). Then the mechanism $L : \mathcal{Y} \to \mathcal{P}(\mathcal{Z})$, described by the pdf D, is called a* Laplace mechanism *from $(\mathcal{Y}, d_{\mathcal{Y}})$ to $\mathcal{Z}$.*

Fact 4.2 ([McSherry 2007])**.** *Any Laplace mechanism from $(\mathcal{Y}, d_{\mathcal{Y}})$ to $\mathcal{Z}$ satisfies $d_{\mathcal{Y}}$-privacy.*

Proof. For the pdf describing the mechanism we have:

$$\begin{aligned} D(y)(z) &= \lambda(z)\, e^{-d_{\mathcal{Y}}(y,z)} \\ &\leq \lambda(z)\, e^{-(d_{\mathcal{Y}}(y',z) - d_{\mathcal{Y}}(y,y'))} && \text{triangle ineq.} \\ &= e^{d_{\mathcal{Y}}(y,y')} \lambda(z)\, e^{-d_{\mathcal{Y}}(y',z)} \\ &= e^{d_{\mathcal{Y}}(y,y')} D(y')(z) \end{aligned}$$

for all $y, y' \in \mathcal{Y}, z \in \mathcal{Z}$. The above inequality can be directly extended from the pdf to the measures, thus we conclude that the mechanism satisfies $d_{\mathcal{Y}}$-privacy. □

Figure 4.2 provides instantiations of the general definition for various choices of $\mathcal{Y}, \mathcal{Z}$ and $d_{\mathcal{Y}}$ used in this thesis, by properly adjusting $\lambda(z)$. The basic case (i) is that of the one-dimensional continuous Laplace mechanism. Similarly, we can define a two-dimensional continuous Laplace mechanism (used in Chapters 5 and 6), measuring the distance between points by either the Euclidean (ii) or the Manhattan (iii) metric. In the discrete setting, we obtain the Truncated Geometric mechanism TG_ϵ [Ghosh 2009], given by (iv), using a quantized set of reals as input. We denote by $q[0..k]$ the set $\{qi \mid i \in 0..k\}$, i.e. the set of $k+1$ quantized reals with step size $q > 0$.

$$
\begin{array}{lllll}
\text{(i)} & \mathcal{Y} \subset \mathbb{R}, \quad \mathcal{Z} = \mathbb{R} & d_{\mathcal{Y}} = \epsilon d_{\mathbb{R}} & \lambda_\epsilon(z) = \dfrac{\epsilon}{2} \\
\text{(ii)} & \mathcal{Y} \subset \mathbb{R}^2, \ \mathcal{Z} = \mathbb{R}^2 & d_{\mathcal{Y}} = \epsilon d_2 & \lambda_\epsilon(z) = \dfrac{\epsilon^2}{2\pi} \\
\text{(iii)} & \mathcal{Y} \subset \mathbb{R}^2, \ \mathcal{Z} = \mathbb{R}^2 & d_{\mathcal{Y}} = \epsilon d_1 & \lambda_\epsilon(z) = \dfrac{\epsilon^2}{4} \\
\text{(iv)} & \mathcal{Y} = \mathcal{Z} = q[0..k] & d_{\mathcal{Y}} = \epsilon d_{\mathbb{R}} & \lambda_\epsilon(z) = \begin{cases} \frac{e^{q\epsilon}}{e^{q\epsilon}+1} & z \in \{0, qk\} \\ \frac{e^{q\epsilon}-1}{e^{q\epsilon}+1} & 0 < z < qk \end{cases}
\end{array}
$$

Figure 4.2: Instantiations of the Laplace mechanism

4.2.2 Mechanisms of Optimal Utility

Answering a query privately is useless if the consumer gets no information about the real answer, thus it is crucial to analyze the mechanism's utility. We consider consumers (e.g. data analysts) applying Bayesian inference to map the mechanism's output to a guess that maximizes their expected gain. A consumer is characterized by a prior π on the set of secrets, and a gain function g (assumed to be monotone w.r.t. a metric of reference, which is always $d_{\mathbb{R}}$ for the needs of this thesis). The utility $\mathcal{U}(H, \pi, g)$ of a mechanism H for such a consumer is given by her expected gain under an optimal remap strategy $r : \mathcal{Z} \to \mathcal{X}$.

$$\mathcal{U}(H, \pi, g) = \max_r \textstyle\sum_{x,z} \pi(x) H(x)(z) g(x, r(z))$$

This is the Bayesian notion of utility [Ghosh 2009], but our results can be extended to risk-averse consumers. It is worth noting that this definition is equivalent to the one of AdvError defined in Section 3.1 and used to quantify the privacy of a location privacy protacion mechanism. In the context of location privacy, the consumer (in this case, the service provider) is not assumed to perform any operation on the reported location when retrieving the results for the user. Hence the utility measure does not depend on any remapping. However, the adversary is assumed to

perform a post-pocessing of the observed location, based on his prior knowledge, in order to get a more accurate estimation of the user's real position. Therefore, the measure used to quantify the privacy of the mechanism (which is considered to be the expected error of the adversary), does take this remapping into account.

A natural question to ask, then, is whether, for a *given query* f, there exists a mechanism that universally (i.e. for all priors and gain functions) provides optimal utility. Let $\mathcal{H}_f(d_\mathcal{X})$ be the set of all mechanisms $H : \mathcal{Y} \to \mathcal{Z}$ (for any $\mathcal{Z}$) such that $H \circ f$ satisfies $d_\mathcal{X}$-privacy. All mechanisms in $\mathcal{H}_f(d_\mathcal{X})$ can be used to answer f privately, hence we are interested in the one that maximizes the expected gain.

Definition 4.7. *A mechanism $H \in \mathcal{H}_f(d_\mathcal{X})$ is f-$d_\mathcal{X}$-optimal iff $\mathcal{U}(H, \pi, g) \geq \mathcal{U}(H', \pi, g)$ for all $H' \in \mathcal{H}_f(d_\mathcal{X})$, all priors π and all gain functions g.*

The existence of (universally) optimal mechanisms is far from trivial. For standard differential privacy, a well-known result from [Ghosh 2009] states that such a mechanism does exist for *counting* queries, i.e. those of the form "how many users satisfy property P".

Theorem 4.4 ([Ghosh 2009])**.** *Let $\mathcal{Y} = [0..k]$ and let $f : \mathcal{V}^n \to \mathcal{Y}$ be a counting query. Then the TG_ϵ mechanism with input $\mathcal{Y}$ is f-ϵd_h-optimal for all $\epsilon > 0$.*

On the other hand, a well-known impossibility result [Brenner 2010] states that counting queries are essentially the only ones for which an optimal mechanism exists. This result is based on the concept of the *induced graph* $\sim_f$ of a query $f : \mathcal{V}^n \to \mathcal{Y}$, defined as: $y \sim_f y'$ iff $\exists x \sim_h x'$ s.t. $f(x) = y, f(x') = y'$.

Theorem 4.5 ([Brenner 2010])**.** *Let $f : \mathcal{V}^n \to \mathcal{Y}$ be a query such that $\sim_f$ is not a path graph. Then no f-ϵd_h-optimal mechanism exists for any $\epsilon < \ln 2$.*

Thus, most interesting queries, e.g. the sum and average, have no optimal mechanisms.

However, the above negative result and the concept of the induced graph are tied to the Hamming metric d_h. This raises the question of whether this special status of counting queries holds for any metric $d_\mathcal{X}$. To answer this question, we will give a sufficient condition for showing the optimality of TG_ϵ for an arbitrary query f and metric $d_\mathcal{X}$, based on the concept of uniform sensitivity.

We start by introducing the concept of a mechanism being optimal not w.r.t. a specific query, but w.r.t. *the metric of its input domain.* Let $\mathcal{H}(d_\mathcal{Y})$ be the set of all mechanisms $H : \mathcal{Y} \to \mathcal{Z}$ (for any $\mathcal{Z}$) satisfying $d_\mathcal{Y}$-privacy.

Definition 4.8. *A mechanism $H \in \mathcal{H}(d_\mathcal{Y})$ is $d_\mathcal{Y}$-optimal iff $\mathcal{U}(H, \pi, g) \geq \mathcal{U}(H', \pi, g)$ for all $H' \in \mathcal{H}(d_\mathcal{Y})$, all priors π and all gain functions g.*

Notice the difference between $d_\mathcal{Y}$-optimal and f-$d_\mathcal{X}$-optimal; the latter refers to a specific query. The two notions can be related in the case of uniformly sensitive queries by the following result:

Proposition 4.2. *Assume that f is uniformly Δ-sensitive w.r.t. $d_{\mathcal{X}}, d_{\mathcal{Y}}$. Then H is f-$\Delta d_{\mathcal{X}}$-optimal iff it is $d_{\mathcal{Y}}$-optimal.*

Proof. From uniform Δ-sensitivity and Theorem 4.3, we get that $\mathcal{H}(d_{\mathcal{Y}}) = \mathcal{H}_f(\Delta d_{\mathcal{X}})$. Then the result follows directly from the definition of optimality (Definitions 4.7,4.8). □

The importance of the induced graph $\sim_f$ in optimality results comes from the fact that f is always uniformly sensitive w.r.t. the metric induced by $\sim_f$.

Proposition 4.3. *Let f be a query with induced graph $\sim_f$, and let d_f be the metric induced by $\sim_f$. Then f is uniformly 1-sensitive w.r.t. d_h, d_f.*

Proof. Let $x, x' \in \mathcal{X}$ and let $n = d_h(x, x')$. We first need to show that f is 1-sensitive w.r.t. d_h, d_f, that is $d_f(f(x), f(x')) \leq n$. Since d_h is induced by $\sim_h$, there exists a $\sim_h$-path $x_1 \ldots, x_n$ such that $x = x_1, x' = x_n$. By definition of $\sim_f$ we have that $f(x_i) \sim_f f(x_{i+1})$, thus $f(x_1), \ldots, f(x_n)$ is a $\sim_f$-path of length n from $f(x)$ to $f(x')$. Since $d_f(f(x), f(x'))$ is the length of the shortest such path, we have that $d_f(f(x), f(x')) \leq n$.

For the "uniformly" part, let $y, y' \in \mathcal{Y}$ and $n = d_f(y, y')$. We need to show that there exists a tight and 1-expansive chain from y to y'.

Since d_f is induced by $\sim_f$, there exist a $\sim_f$-path $\tilde{y} = y_1, \ldots, y_n$ such that $y = y_1, y' = y_n$ This implies that $d_f(y_i, y_{i+1}) = 1$ and thus $d_f(\tilde{y}) = n = d_f(y, y')$ so the chain is tight.

Moreover, from the definition of $\sim_f$ we have that there exist $x \sim_h x'$ such that $f(x) = y_i, f(x') = y_{i+1}$, so $d_h(x, x') = 1$ which means that y_i, y_{i+1} are 1-expansive, and this happens for all $1 \leq i < n$ so the chain is 1-expansive. □

We can now show the optimality of TG_ϵ with input $q[0..k]$ w.r.t. the $\epsilon d_{\mathbb{R}}$ metric, independently from any query.

Proposition 4.4. *Let $\mathcal{Y}_q = q[0..k]$. The TG_ϵ mechanism with input $\mathcal{Y}_q$ is $\epsilon d_{\mathbb{R}}$-optimal for all $\epsilon > 0$.*

Proof. Fix $\mathcal{Y} = 0..k$ and $\mathcal{Y}_q = q[0..k]$ for some $k \in \mathbb{N}, q > 0$, and let $TG_\epsilon(\mathcal{Y}), TG_\epsilon(\mathcal{Y}_q)$ denote the Truncated Geometric mechanisms with input $\mathcal{Y}, \mathcal{Y}_q$ respectively.

From Theorem 4.4 we known that $TG_\epsilon(\mathcal{Y})$ is f-ϵd_h-optimal when f is a counting query. For counting queries, d_f (the metric that corresponds to their induced graph) and $d_{\mathbb{R}}$ coincide, thus from Prop 4.3 we get that f is uniformly 1-sensitive w.r.t. $d_h, d_{\mathbb{R}}$. Then from Prop 4.2 we have that that $TG_\epsilon(\mathcal{Y})$ is $\epsilon d_{\mathbb{R}}$-optimal. This mechanism has pdf

$$D_\epsilon(y)(z) = \lambda_\epsilon(z)\, e^{-\epsilon\, d_{\mathbb{R}}(y,z)} \quad \lambda_\epsilon(z) = \begin{cases} \frac{e^\epsilon}{e^\epsilon+1} & z \in \{0, k\} \\ \frac{e^\epsilon-1}{e^\epsilon+1} & 0 < z < k \end{cases}$$

We now show that $TG_\epsilon(\mathcal{Y}_q)$ is also $\epsilon d_\mathbb{R}$-optimal. The metric spaces $(\mathcal{Y}, \epsilon d_\mathbb{R})$ and $(\mathcal{Y}_q, \epsilon\frac{1}{q}d_\mathbb{R})$ are isometric. So we can obtain a mechanism D'_ϵ with input $\mathcal{Y}_q$ by replacing i with qi and $\epsilon d_\mathbb{R}$ with $\epsilon\frac{1}{q}d_\mathbb{R}$). The pdf of this mechanism is:

$$D'_\epsilon(y)(z) = \lambda_\epsilon(z)\, e^{-\epsilon\,\frac{1}{q}d_\mathbb{R}(y,z)} \quad \lambda_\epsilon(z) = \begin{cases} \frac{e^\epsilon}{e^\epsilon+1} & z \in \{0, qk\} \\ \frac{e^\epsilon-1}{e^\epsilon+1} & 0 < z < k \end{cases}$$

Due to the isometry, D_ϵ satisfies $\epsilon d_\mathbb{R}$-privacy iff D'_ϵ satisfies $\epsilon\frac{1}{q}d_\mathbb{R}$-privacy, thus (from the optimality of D_ϵ) it follows that D'_ϵ is $\epsilon\frac{1}{q}d_\mathbb{R}$-optimal for all $\epsilon > 0$.

Finally we define:

$$D''_\epsilon = D'_{q\epsilon}$$

From the optimality of D'_ϵ we get that D''_ϵ is $(q\epsilon)\frac{1}{q}d_\mathbb{R}$-optimal, i.e. it is $\epsilon d_\mathbb{R}$-optimal.

This concludes the proof, since D''_ϵ is exactly the pdf of $TG_\epsilon(\mathcal{Y}_q)$. □

The results above bring us directly to our sufficient condition.

Theorem 4.6. *Let $\mathcal{Y} = q[0..k]$ and assume that $f : \mathcal{X} \to \mathcal{Y}$ is uniformly Δ-sensitive w.r.t. $d_\mathcal{X}, d_\mathbb{R}$. Then the TG_ϵ mechanism with input $\mathcal{Y}$ is f-$\Delta d_\mathcal{X}$-optimal.*

Proof. Direct corollary of Prop 4.4 and Prop 4.2. □

In the following sections we show that this condition is indeed satisfied by several important queries, including the sum and average, for various metrics of interest.

4.3 Privacy in Statistical Databases

In this section, we investigate privacy notions in the context of statistical databases, other than the standard differential privacy. In contrast to the Hamming distance, which can be defined independently from the structure of the universe $\mathcal{V}$, we are interested in metrics that depend on the actual values and the distance between them. To this end, we assume that the universe is equipped with a metric $d_\mathcal{V}$, measuring how far apart two values are. When the universe is numeric (i.e. $\mathcal{V} \subset \mathbb{R}$) then $d_\mathcal{V} = d_\mathbb{R}$ is the natural choice. In the case of null values, we can extend a metric $d_\mathcal{V}$ from $\mathcal{V}$ to $\mathcal{V}_\varnothing$ by considering $\varnothing$ to be maximally distant from all other values, that is taking $d_\mathcal{V}(\varnothing, v) = d_\mathcal{V}(\mathcal{V}), v \in \mathcal{V}$. Note that this construction preserves the maximum distance between values, i.e. $d_\mathcal{V}(\mathcal{V}_\varnothing) = d_\mathcal{V}(\mathcal{V})$.

The first metric we consider, the normalized Manhattan metric, allows to strengthen differential privacy, obtaining a notion that not only protects the value of an individual, but also offers higher protection to small modifications of a value. Then we relax this metric, to obtain a weaker notion, that only protects the "accuracy" of an individual's value, but offers higher utility.

4.3.1 The Normalized Manhattan Metric

Differential privacy provides indistinguishability between databases differing in a single individual, but the level of distinguishability is independent from the actual value in those databases. Consider for example a database with salary information, and two adjacent databases $x \sim_i x'$ ($\sim_i$ denoting that they differ only in the value of the i-th individual) with $x[i] = v, x'[i] = v'$. A differentially private mechanism offers distinguishability level $\boldsymbol{\varepsilon}(x, x') = \epsilon$, independently from v, v'. This means that when $v = 0, v' = 1\text{M}$, the indistinguishability level between x, x' will be the same as in the case $v = 20.000$, $v' = 20.001$.

One might expect, however, to have better protection in the second case, since the change in the individual's data is insignificant. Being insensitive to such small changes seems a reasonable privacy requirement since many queries (e.g. sum, average, etc) are themselves insensitive to small perturbations. The equal treatment of values is particularly problematic when we are obliged to use a "weak" ϵ, due to a high sensitivity. In this case, all values are only guaranteed to be weakly protected, while we could expect that at least close values would still enjoy high protection.

The normalized Manhattan metric $\widetilde{d_1}$ expresses exactly this idea. Databases differing in a single value have distance at most 1, but the distance can be substantially smaller for small modifications of values, offering higher protection in those cases. The Manhattan metric d_1 on $\mathcal{V}^n$ and its normalized version $\widetilde{d_1}$ are defined as:[1] $d_1(x, x') = \sum_{i=1}^{n} d_{\mathcal{V}}(x[i], x'[i])$ and $\widetilde{d_1}(x, x') = \frac{d_1(x,x')}{d_{\mathcal{V}}(\mathcal{V})}$. Similarly to differential privacy, we use a scaled version $\epsilon\widetilde{d_1}$ of the metric, to properly adjust the distinguishability level.

Concerning the operational characterizations of Section 4.1.2, the hiding functions and neighborhoods suitable for this metric are:

$$\phi_{i,w} = x[{}^{w(x[i])}/_{i}] \text{ for } w : \mathcal{V} \to \mathcal{V} \qquad N_{i,V}(x) = \{x[{}^{v}/_{i}] \mid v \in V\}$$
$$\Phi_1 = \{\phi_{i,w} \mid i \in 1..n, w : \mathcal{V} \to \mathcal{V}\} \qquad \mathcal{N}_1 = \{N_{i,V}(x) \mid x \in \mathcal{V}^n, i \in 1..n, V \subseteq \mathcal{V}\}$$

A hiding function $\phi_{i,w}$ replaces the value of individual i by applying an arbitrary substitution of values w (instead of replacing with a fixed value as $\phi_{i,v}$ does). Moreover, for the adversary, knowing $N_{i,V}(x)$ means that he knows the values of all individuals in the database but i, and that the value of i lies within V. Note that $\Phi_h \subset \Phi_1$ and $\mathcal{N}_h \subset \mathcal{N}_1$. We show that $\Phi_1, \mathcal{N}_1$ are "canonical".

Proposition 4.5. *$\Phi_1, \mathcal{N}_1$ are maximally tight w.r.t. both $d_1, \widetilde{d_1}$.*

Proof. We first consider d_1. Let $x, x' \in \mathcal{V}^n$, we show that there exist a tight chain from x to x' that is both a maximal Φ_1-chain and a maximal $\mathcal{N}_1$-chain. We recursively create a chain $x_1, \ldots, x_{n+1}$ from x to x' by modifying one element at a

[1] Note that in the differential privacy literature, the d_1 distance is often used on *histograms*. This metric is closely related to the standard d_h distance on $\mathcal{V}^n$ (it depends only on the record counts), and different than d_1 on $\mathcal{V}^n$ which depends on the actual values.

time:

$$x_1 = x$$
$$x_{i+1} = x_i[{}^{x'[i]}/_i] \quad i \in 1..n$$

It is easy to see that $d_1(x, x') = \sum_{i=1}^{n} d_1(x_i, x_{i+1})$ so the chain is tight w.r.t. d_1.

Fix any $i \in 1..n$ and let $w : \mathcal{V} \to \mathcal{V}$ defined as:

$$w(v) = \begin{cases} x'[i] & \text{if } v = x[i] \\ x[i] & \text{if } v = x'[i] \\ v & \text{otherwise} \end{cases}$$

For the hiding function $\phi_{i,w} \in \Phi_1$ it holds that

$$\phi_{i,w}(x_i) = x_{i+1} \qquad \phi_{i,w}(x_{i+1}) = x_i \qquad d_1(x_i, x_{i+1}) = d_1(\phi_{i,w})$$

hence the chain is a maximal Φ_1-chain.

Moreover, let $V = \{x[i], x'[i]\}$. For the neighborhood $N_{i,V}(x_i) \in \mathcal{N}_1$ it holds that

$$\{x_i, x_{i+1}\} \subseteq N_{i,V}(x_i) \qquad d_1(x_i, x_{i+1}) = d_1(N_{i,V}(x_i))$$

so the chain is a maximal $\mathcal{N}_1$-chain.

The case of $\widetilde{d}_1$ is similar, since it is a scaled version of d_1. □

From Theorem 4.1, we conclude that $\epsilon\widetilde{d}_1$-privacy is equivalent to requiring that the adversary's posterior distributions with or without hiding i's value should be at most $2\epsilon\widetilde{d}_1(\phi_{i,w})$ distant. Since $\widetilde{d}_1(\phi_{i,w}) \leq 1$, hiding the individual's value in any way has small effect on the adversary's conclusions. But if i's value is replaced by one close to it, $\widetilde{d}_1(\phi_{i,w})$ can be much lower than 1, meaning that the effect on the adversary's conclusions is even smaller.

Then, from Theorem 4.2 we conclude that $\epsilon\widetilde{d}_1$-privacy is equivalent to requiring that, for an informed adversary knowing the value of all individuals but i, and moreover knowing that i's value lies in V, his conclusions differ from his initial knowledge by at most $\epsilon\frac{d_\mathcal{V}(V)}{d_\mathcal{V}(\mathcal{V})}$. This difference is at most ϵ, but can be much smaller if values in V are close to each other, meaning that for an adversary who knows i's value with high accuracy, the gain is even smaller.

Intuitively, $\epsilon\widetilde{d}_1$-privacy offers a stronger notion of privacy than ϵd_h-privacy:

Proposition 4.6. *$\widetilde{d}_1 \leq d_h$, thus $\epsilon\widetilde{d}_1$-privacy implies ϵd_h-privacy.*

Proof. Fix $x, x' \in \mathcal{V}^n$ and let $I = \{i \in 1..n \mid x[i] \neq x[i']\}$. Then

$$\widetilde{d}_1(x, x') = \frac{\sum_{i \in I} d_\mathcal{V}(x[i], x[i'])}{d_\mathcal{V}(\mathcal{V})} \leq \frac{\sum_{i \in I} d_\mathcal{V}(\mathcal{V})}{d_\mathcal{V}(\mathcal{V})} = |I| = d_h(x, x')$$

□

We continue by introducing $\mathcal{N}$-tightness, a relaxed version of the concept of maximal $\mathcal{N}$-tightness (Def 4.3), by simply dropping the requirement $d_{\mathcal{X}}(x_i, x_{i+1}) = d_{\mathcal{X}}(N)$ from Def 4.3.

Definition 4.9. *Let $\mathcal{N} \subseteq 2^{\mathcal{X}}$. A chain $\tilde{x}$ is called an* $\mathcal{N}$-chain *iff for every step i there exists $N \in \mathcal{N}$ such that $\{x_i, x_{i+1}\} \subseteq N$. Then $\mathcal{N}$ is called* tight *w.r.t. $d_{\mathcal{X}}$ iff $\forall x, x' \in \mathcal{X}$ there exists a tight $\mathcal{N}$-chain from x to x'.*

We can now show the (maximal w.r.t. d_h, simple w.r.t. $d_1, \widetilde{d}_1$) tightness of $\mathcal{N}_h$, which will be useful later on.

Proposition 4.7. *$\mathcal{N}_h$ is maximally tight w.r.t. d_h and tight w.r.t. both $d_1, \widetilde{d}_1$.*

Proof. Let $x, x' \in \mathcal{V}^n$. We need to show that there exists a tight (also maximal in the case of d_h) $\mathcal{N}_h$-chain from x to x'. We recursively create a chain $x_1, \ldots, x_{n+1}$ from x to x' by modifying one element at a time:

$$\begin{aligned} x_1 &= x \\ x_{i+1} &= x_i[{}^{x'[i]}/_i] \quad i \in 1..n \end{aligned}$$

It is easy to see that $d(x, x') = \sum_{i=1}^{n} d(x_i, x_{i+1})$, for all $d \in \{d_h, d_1, \widetilde{d}_1\}$, so the chain is tight w.r.t. all three metrics. Moreover, we have that $\{x_i, x_{i+1}\} \subseteq N_i(x_i)$ so the chain is an $\mathcal{N}_h$-chain w.r.t. all metrics.

For d_h, it also holds that $d_h(x_i, x_{i+1}) = d_h(N_i(x_i)) = 1$, so the chain is a maximal $\mathcal{N}_h$-chain w.r.t. d_h. □

We continue with a lemma that facilitates proofs of Δ-sensitivity by reducing the pairs of secrets x, x' that one needs to check.

Lemma 4.2. *Let $\mathcal{N} \subseteq 2^{\mathcal{X}}$ be tight (Def 4.9) and assume:*

$$d_{\mathcal{Y}}(f(x), f(x')) \leq \Delta d_{\mathcal{X}}(x, x') \qquad \forall N \in \mathcal{N}, x, x' \in N$$

Then f is Δ-sensitive w.r.t. $d_{\mathcal{X}}, d_{\mathcal{Y}}$.

Proof. Fix $x, x' \in \mathcal{X}$, we need to show that $d_{\mathcal{Y}}(f(x), f(x')) \leq \Delta d_{\mathcal{X}}(x, x')$. Since $\mathcal{N}$ is tight there exist a tight $\mathcal{N}$-chain $x = x_1, \ldots, x_n = x'$, such that each step x_i, x_{i+1} belongs to some set $N \in \mathcal{N}$. We have:

$$\begin{aligned} & d_{\mathcal{Y}}(f(x), f(x')) \\ & \leq \sum_{i=1}^{n-1} d_{\mathcal{Y}}(f(x_i), f(x_{i+1})) && \text{triangle ineq.} \\ & \leq \Delta \sum_{i=1}^{n-1} d_{\mathcal{X}}(x_i, x_{i+1}) && \text{hypoth., } x_i, x_{i+1} \in N \\ & = \Delta d_{\mathcal{X}}(x, x') && \text{tightness of chain} \end{aligned}$$

□

By Proposition 4.6, since distances in $\widetilde{d}_1$ can be smaller than those in d_h, the sensitivity of a query w.r.t. $\widetilde{d}_1$ is in general greater than the sensitivity w.r.t. d_h, which means that to achieve $\epsilon\widetilde{d}_1$-privacy we need to apply more noise. However, for a general class of queries, it turns out that the two sensitivities coincide.

Definition 4.10. *A query f belongs to the family $\mathcal{C}$ iff $d_{\mathbb{R}}(f(x), f(x')) \leq d_{\mathcal{V}}(x[i], x'[i])$ for all $i \in 1..n$, $x \sim_i x' \in \mathcal{V}^n$, and moreover $\exists x \sim_i x' \in \mathcal{V}^n$ such that $d_{\mathbb{R}}(f(x), f(x')) = d_{\mathcal{V}}(\mathcal{V})$.*

Proposition 4.8. *Let $f \in \mathcal{C}$. The sensitivity of f w.r.t. both $d_h, d_{\mathbb{R}}$ and $\widetilde{d}_1, d_{\mathbb{R}}$ is $d_{\mathcal{V}}(\mathcal{V})$.*

Proof. Let $f \in \mathcal{C}$. We first show that f is $d_{\mathcal{V}}(\mathcal{V})$-sensitive w.r.t. both $d_h, d_{\mathbb{R}}$ and $\widetilde{d}_1, d_{\mathbb{R}}$. From Prop 4.7 together with Lemma 4.2, we only need to show the sensitivity for databases x, x' from some set $N_i(x) \in \mathcal{N}_h$, i.e. $x \sim_i x'$.

For d_h, we have $d_h(x, x') = 1$, thus

$$d_{\mathcal{Y}}(f(x), f(x')) \leq d_{\mathcal{V}}(x[i], x'[i]) \leq d_{\mathcal{V}}(\mathcal{V}) d_h(x, x')$$

For $\widetilde{d}_1$ we have:

$$d_{\mathcal{Y}}(f(x), f(x')) \leq d_{\mathcal{V}}(x[i], x'[i]) = d_{\mathcal{V}}(\mathcal{V}) \widetilde{d}_1(x, x')$$

Then, for any $\Delta < d_{\mathcal{V}}(\mathcal{V})$, f is not Δ-sensitive for neither metric, since from Def 4.10 there exists $x \sim_i x'$ such that

$$d_{\mathcal{Y}}(f(x), f(x')) = d_{\mathcal{V}}(\mathcal{V}) > \Delta d_h(x, x')$$

and similarly for $\widetilde{d}_1$. □

Intuitively, the class $\mathcal{C}$ contains queries for which the sensitivity is obtained for values that are maximally distant. For those queries, using the Truncated Geometric mechanism we can achieve a notion of privacy stronger than differential privacy *using the same amount of noise*!

Results About Some Common Queries

We now focus on some commonly used queries, namely the sum, average and p-percentile queries. Note that other commonly used queries such as the max, min and median queries are specific cases of the p-percentile query. In the following, we assume that the universe is $\mathcal{V} = q[0..k]_{\varnothing}$ with metric $d_{\mathbb{R}}$, and take $\mathcal{X} = \mathcal{V}^n \setminus \{\langle \varnothing, \ldots, \varnothing \rangle\}$, that is we exclude the empty database so that the queries can be always defined.

For these queries we obtain two results: first, we show that they belong to the $\mathcal{C}$ family, which means that we can achieve $\epsilon\widetilde{d}_1$-privacy via the TG_ϵ mechanism, using the same amount of noise that we would need for standard differential privacy.

Proposition 4.9. *The* sum, avg, p-perc *queries belong to $\mathcal{C}$.*

Proof. The universe is assumed to be $\mathcal{V} = q[0..k]_\varnothing$ for some $k \in \mathbb{N}, q > 0$. Let $x \sim_i x' \in \mathcal{V}^n$. We first show that $d_\mathbb{R}(f(x), f(x')) \leq d_\mathcal{V}(x[i], x'[i])$.

For sum, it is easy to see that

$$|\text{sum}(x) - \text{sum}(x')| = \begin{cases} d_\mathcal{V}(x[i], x'[i]) & x[i] \neq \varnothing, x'[i] \neq \varnothing \\ x'[i] & x[i] = \varnothing \\ x[i] & x'[i] = \varnothing \end{cases}$$

Note that $x'[i] \leq d_\mathcal{V}(x[i], x'[i]) = qr$ in the case $x[i] = \varnothing$ (and similarly for $x'[i] = \varnothing$).[2]

Consider now $f \in \{\text{avg}, p\text{-perc}\}$. From Theorem 4.9 (which will be stated and proved in upcoming Section 4.4) we know that both queries are 1-sensitive w.r.t. $d_\infty, d_\mathbb{R}$. And since $d_\infty(x, x') = d_\mathcal{V}(x[i], x'[i])$ we have:

$$d_\mathbb{R}(f(x), f(x')) \leq d_\infty(x, x') = d_\mathcal{V}(x[i], x'[i])$$

Finally, we need to show that there exist $x \sim_i x' \in \mathcal{V}^n$ such that $d_\mathbb{R}(f(x), f(x')) = d_\mathcal{V}(\mathcal{V}) = qk$. We construct $x = \langle 0, \varnothing, \ldots, \varnothing \rangle$, $x' = \langle qk, \varnothing, \ldots, \varnothing \rangle$. These databases satisfy $d_\mathbb{R}(f(x), f(x')) = qk$ for all queries. □

More interestingly, we can show that the Truncated Geometric mechanism is in fact universally optimal w.r.t. $\widetilde{d}_1$ for such queries.

Theorem 4.7. *The* sum, avg *and* p-perc *queries are all uniformly* qk*-sensitive w.r.t.* $\widetilde{d}_1, d_\mathbb{R}$.

Proof. First we have to show that the queries are qk-sensitive w.r.t. $\widetilde{d}_1, d_\mathbb{R}$. This comes from Prop 4.8 and 4.9, since all queries belong to the family $\mathcal{C}$.

We now show the uniform sensitivity of sum. Let $y, y' \in q[0..nk]$ and assume that $y \geq y'$. It is easy to see that we can construct databases x, x' such that $\text{sum}(x) = y, \text{sum}(x') = y'$ and $x[i] \geq x'[i]$ for all $i \in 1..n$. For x, x' we have

$$\begin{aligned} d_1(x, x') &= \textstyle\sum_i |x[i] - x'[i]| \\ &= |\textstyle\sum_i x[i]| - |\sum_i x'[i]| && x[i] \geq x'[i] \\ &= d_\mathbb{R}(\text{sum}(x), \text{sum}(x')) \end{aligned}$$

Thus $d_\mathbb{R}(y, y') = qk\ \widetilde{d}_1(x, x')$, which means that the chain y, y' is qk-expansive w.r.t. $\widetilde{d}_1$.

Finally, for $f \in \{\text{avg}, p\text{-perc}\}$ let $y, y' \in q[0..k]$. We construct two databases x, x' with a single present individual as follows:

$$x = \langle y, \varnothing, \ldots, \varnothing \rangle \qquad x' = \langle y', \varnothing, \ldots, \varnothing \rangle$$

It is easy to see that $f(x) = y, f(x') = y'$ and $d_1(x, x') = d_\mathbb{R}(y, y')$. Thus $d_\mathbb{R}(y, y') = qk\ \widetilde{d}_1(x, x')$ which means that the chain y, y' is qk-expansive w.r.t. $\widetilde{d}_1$. □

Corollary. *$TG_{\epsilon/qk}$ is f-$\epsilon\widetilde{d}_1$-optimal for $f \in \{\text{sum}, \text{avg}, p\text{-perc}\}$, $\epsilon > 0$.*

[2] It is crucial here that $\mathcal{V}$ contains 0, so that $v \leq d_\mathcal{V}(\mathcal{V})$ for all non-null v. If $0 \notin \mathcal{V}$, we can achieve a similar result for sum by adapting the way $d_\mathcal{V}$ treats $\varnothing$.

4.3.2 The Manhattan Metric

In the previous section, we used the normalized Manhattan metric $\epsilon\widetilde{d_1}$, obtaining a strong privacy notion that protects an individual's value, while offering even stronger protection for small changes in an individual's value. This however, requires at least as much noise as standard differential privacy.

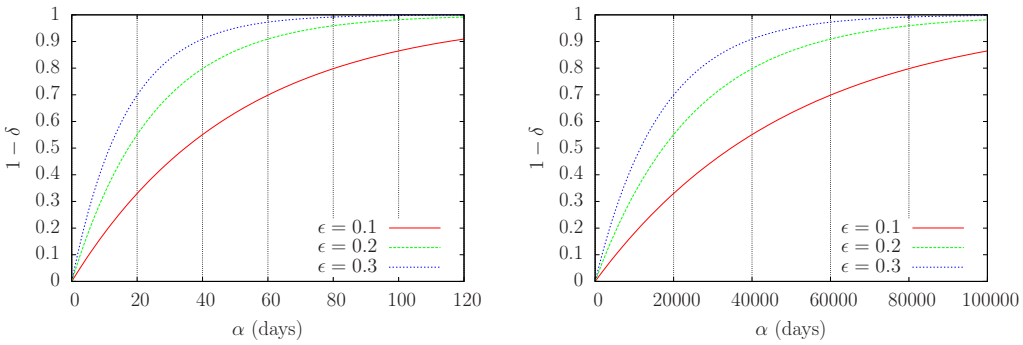
On the other hand, there are applications in which a complete protection of an individual's value is not required. This happens, for instance, in situations when the actual value is not sensitive, but knowing it with high accuracy might allow an adversary to identify the individual. Consider for example a database with the individuals' birthday, or the registration date and time to some social network. This information, by itself, might not be considered private, however knowing such information with minute-accuracy could easily allow to identify an individual. In such situations we might wish to protect only the accuracy of the value, thus achieving privacy with less noise and offering more accurate results.

This can be achieved by the Manhattan metric ϵd_1 (without normalization). This metric might assign a level of distinguishability higher than ϵ for adjacent databases, thus the privacy guarantees could be weaker than those of ϵ-differential privacy. However, adjacent databases with small changes in value will be highly protected, thus an adversary cannot infer an individual's value with accuracy.

Similarly to the previous section, we can obtain characterizations of ϵd_1-privacy using the same hiding functions Φ_1 and neighborhoods $\mathcal{N}_1$. The only difference is that $\epsilon d_1(\phi_{i,w})$ and $\epsilon d_1(N_{i,V})$ can be now higher than ϵ, offering weaker protection. However, when the adversary already knows i's value with high accuracy, meaning that values in V are close to each other, it is guaranteed that his knowledge will increase by a small factor (possibly even smaller than ϵ), ensuring that he cannot infer the value with even higher accuracy.

Note that the sensitivity of a query can be substantially lower w.r.t. d_1 than w.r.t. d_h. For example, the sum query is 1-sensitive w.r.t. d_1 but qr-sensitive w.r.t. d_h. This means that the noise we need to add could be substantially lower, offering better utility at the expense of lower privacy, but still sufficient for a given application.

Example 4.1. *Consider a database containing the registration date on some social network, expressed as the number of days since Jan 1, 2000. We want to privately release the earliest registration date among individuals satisfying some criteria. A registration date itself is not considered sensitive, however from the result of the query it should be impossible to infer whether a particular individual belongs to that set. Since values can range between* 0 *and approximately* 5.000*, the sensitivity of the min query w.r.t.* d_h *is* 5.000*, while w.r.t.* d_1 *it is only* 1*. By using* ϵd_h *we protect (up to the intended level* ϵ*) an individual's registration date within the whole range of values, while by using* $\frac{\epsilon}{5}d_1$ *we provide the intended protection only within a radius of 5 days. More precisely: in the first case two adjacent databases will always have distinguishability level* ϵ*, while in the second case such level of protection is guaranteed only if the individual's registration date differs by at most 5 days in the two databases*



(a) $\frac{\epsilon}{5}d_1$-privacy (b) ϵd_h-privacy

Figure 4.3: Utility for various values of ϵ

(if they differ more the distinguishability level will increase proportionally). The second case, of course, offers less privacy, but, depending on the application, confusion within 5 days can be enough to prevent an individual from being identified. On the other hand, the trade-off with utility can be much more favorable in the second case: In Figure 4.3 we show the utility of a Laplace mechanism for both metrics, in terms of (α, δ)-usefulness (meaning that the mechanism reports a result within distance α from the real value with probability at least $1 - \delta$).[3] *Clearly, $\frac{\epsilon}{5}d_1$-privacy gives acceptable utility while ϵd_h-privacy renders the result almost useless.*

Finally, the optimality result from the previous section also holds for d_1.

Theorem 4.8. *The* sum, avg *and* p-perc *queries are all uniformly* 1*-sensitive w.r.t.* $d_1, d_{\mathbb{R}}$.

Proof. Direct consequence of Theorem 4.7, since $d_1 = qk\ \widetilde{d}_1$. □

Corollary. *TG_ϵ is f-ϵd_1-optimal for $f \in \{\text{sum}, \text{avg}, p\text{-perc}\}$, $\epsilon > 0$.*

4.4 Privacy in Other Contexts: Smart Meters

A smart meter is a device that records the consumption of electrical energy at potentially very short time intervals, and transmits the information to the utility provider, thus offering him the capability to monitor consumption accurately and almost in real-time.

The Problem

Although smart meters can help improving energy management, they create serious privacy threats: By analyzing accurate consumption data, thanks to appliance signature libraries it is possible to identify which electric devices are being

[3] Using Bayesian utility leads to similar results.

used [Lam 2007]. It has even been shown that, depending on the granularity of measurement and the resolution of data, it is possible to deduce what TV channels, and which movies are being watched [Greveler 2012].

Several papers addressed the privacy problems of smart metering in the recent past. The solution proposed in [Danezis 2011] is based on the use of techniques of (standard) differential privacy in order to send sanitized sums of the readings over some period of time (e.g. an hour, a day, a month) to the service provider. Since this solution is tailored to the use of smart metering for billing purposes, the noise added is assumed to be positive.

The Model

For the sake of generality, we assume here that the noise could be of any kind (not necessarily positive). We can regard the readings over the period $[1..n]$ as a tuple $x \in \mathcal{V}^n$, so that $x[i]$ represents the reading at the time i. Since [Danezis 2011] uses the standard differential privacy framework, the distinguishability metric on these tuples is assumed to be the Hamming distance, and therefore the privacy mechanism is tuned to protect the value of $x[i]$, regardless of whether the variation of this value is small or large. However, the solution proposed in [Danezis 2011] is general and can be adapted to a different distinguishability metric.

We argue that for the case of smart meters, the problem that derives from the extreme accuracy of the readings can be addressed with limited noise by adopting a metric that is sensitive also to the distance between values, and not only to the change of the value for a reading $x[i]$. The reason is the same as illustrated in previous section: if we want to protect small variations in the reading of $x[i]$, it is not a good idea to tune the sensitivity on the difference between the extremes values, because we would end up introducing a lot of noise. In fact, the experiments in [Greveler 2012] are performed on actual smart meters that are in the process of being deployed. These meters send readings to the service provider every 2 seconds. The solution proposed in [Danezis 2011] offers good privacy guarantees by completely protecting each measurement. However, such a definition is too strong if reporting values at short intervals is a requirement. With standard differential privacy, we cannot hope to fully protect each measurement without introducing too much noise. On the other hand, using a more relaxed metric, we can at least provide a meaningful privacy guarantee by protecting the accuracy of the values. Some privacy will still be lost, but the attacks described above where the individual's behaviour is completely disclosed, will be prevented.

The Manhattan distance d_1 on $\mathcal{V}^n$, however, is not suitable to model the privacy problem we have here: in fact d_1 is suitable to protect an individual $x[i]$ and its value, while here we want to protect *all the values at the same time*. This is because the adversary, i.e., the service provider, already knows *an approximation of all values*. Note the difference from the case of Section 4.3: there, the canonical adversary knows all exact values except $x[i]$, and for $x[i]$ he only knows an approximate value.

(In the case of standard differential privacy, the canonical adversary knows all values except $x[i]$, and for $x[i]$ he does not even know an approximate value.)

The suitable distance, in this case, is the maximum distance between components, d_∞. In fact, we should consider x, x' "indistinguishable enough" (i.e. $d(x, x') \leq \delta$, for a certain δ) if and only if for each component i, $x[i], x'[i]$ are "indistinguishable enough" (i.e. $d(x[i], x'[i]) \leq \delta$, for the same δ). It is easy to see that the only distance that satisfies this property is $d(x, x') = d_\infty(x, x') = \max_i d_\mathcal{V}(x[i], x'[i])$.

Example 4.2. *We illustrate the application our method to distort the digital signature of a tv program. The grey line in Fig. 4.4(a) represents the energy consumption of the first 5 minutes of Star Trek 11 [Lam 2007]. The black line is (the approximation of) the signature produced by a smart meter that reports the true readings every 10 seconds (the samples are represented by the dots). The blue and the magenta dots in 4.4(b) are obtained by adding laplacian noise to the true readings, with ϵ values .1 and .5 respectively. As we can see, especially in the case of $\epsilon = .5$, the signature is not recognizable.*

Concerning the characterization results, we use hiding functions substituting the value of all readings. Moreover, we use neighborhoods modelling an adversary that knows all readings with some accuracy, i.e. knows that each reading i lies within V_i.

$$\begin{aligned}\Phi_\infty &= \{\phi_{1,w_1} \circ \ldots \circ \phi_{n,w_n} \mid w_i : \mathcal{V} \to \mathcal{V}\ \forall i \in 1..n\}\\ N_{\{V_i\}} &= \{\langle v_1, \ldots, v_n\rangle \mid v_i \in V_i, i \in 1..n\}\\ \mathcal{N}_\infty &= \{N_{\{V_i\}} \mid V_i \subseteq \mathcal{V}, i \in 1..n\}\end{aligned}$$

We can show that $\Phi_\infty, \mathcal{N}_\infty$ are maximally tight.

Proposition 4.10. *$\Phi_\infty, \mathcal{N}_\infty$ are maximally tight w.r.t. d_∞.*

Proof. Let $x, x' \in \mathcal{V}^n$, and consider the trivial 1-step chain x, x'. This chain is trivially tight, we need to show that it is both a maximal Φ_∞-chain and a maximal $\mathcal{N}_\infty$-chain.

For each $i \in 1..n$ we define a function $w_i : \mathcal{V} \to \mathcal{V}$ as:

$$w_i(v) = \begin{cases} x'[i] & \text{if } v = x[i] \\ x[i] & \text{if } v = x'[i] \\ v & \text{otherwise} \end{cases}$$

For the hiding function $\phi = \phi_{1,w_1} \circ \ldots \circ \phi_{n,w_n}$ we have that $\phi \in \Phi_\infty$ and moreover:

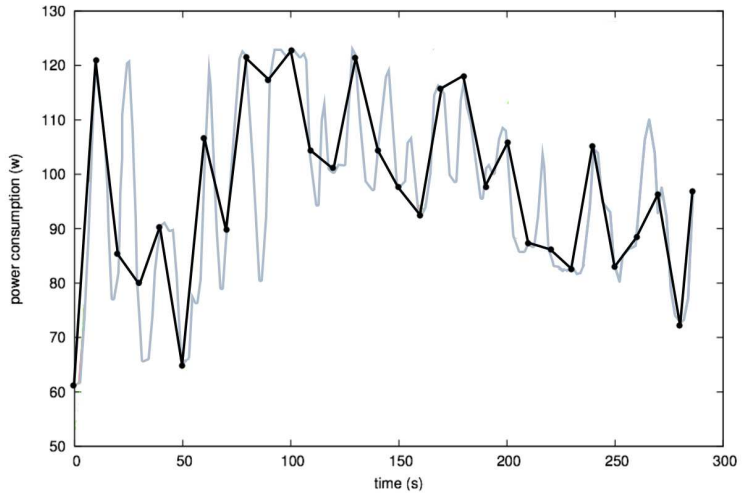
$$\phi(x) = x' \qquad \phi(x') = x \qquad d_\infty(x, x') = d_\infty(\phi)$$

hence the chain is a maximal Φ_∞-chain.

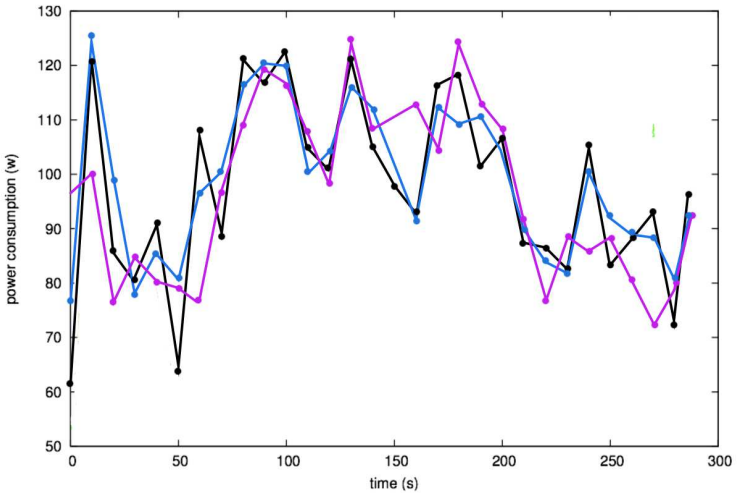
Moreover, let $V_i = \{x[i], x'[i]\}, i \in 1..n$. For the neighborhood $N_{\{V_i\}} \in \mathcal{N}_\infty$ it holds that

$$\{x, x'\} \subseteq N_{\{V_i\}} \qquad d_\infty(x, x') = d_\infty(N_{\{V_i\}})$$

so the chain is a maximal $\mathcal{N}_\infty$-chain. □



(a)



(b)

Figure 4.4: Digital signature of a tv program (a) and its noisy reporting (b).

Finally, we show that TG_ϵ is universally optimal for avg and p-perc.

Theorem 4.9. *The* avg *and* p-perc *queries are both uniformly* 1*-sensitive w.r.t.* $d_\infty, d_\mathbb{R}$.

Proof. The universe is assumed to be $\mathcal{V} = q[0..k]_\varnothing$ for some $k \in \mathbb{N}, q > 0$. The p-percentile query ($0 \leq p < 100$) is defined as $p\text{-perc}(x) = \text{sort}(x)[l]$ for $l = \lfloor \frac{p}{100} m + 1 \rfloor$, where m is the number of non-null values in x and sort returns a sorted version of x (after removing the null values). We also define $I_\varnothing(x) = \{i \in 1..n \mid x[i] = \varnothing\}$.

We fist show that both queries are 1-sensitive w.r.t. $d_\infty, d_\mathbb{R}$. Let $x, x' \in \mathcal{V}^n$. If $I_\varnothing(x) \neq I_\varnothing(x')$ then x, x' are maximally distant, i.e. $d_\infty(x, x') = d_\infty(\mathcal{V}^n) = qk$.

Then for both queries it trivially holds that $d_{\mathbb{R}}(f(x), f(x')) \leq qk = d_\infty(x, x')$ since their range is $q[0..k]$.

It remains to show 1-sensitivity for the case $I_\varnothing(x) = I_\varnothing(x') = I$. For the average query we have

$$\begin{aligned}
&d_{\mathbb{R}}(\text{avg}(x), \text{avg}(x')) \\
&= \frac{1}{|I|}|(\textstyle\sum_{i\in I} x[i]) - (\sum_{i\in I} x'[i])| \\
&= \frac{1}{|I|}|\textstyle\sum_{i\in I} (x[i] - x'[i])| \\
&\leq \frac{1}{|I|}\textstyle\sum_{i\in I} |x[i] - x'[i]| && \text{subadditivity of } |\cdot| \\
&\leq \frac{1}{|I|}\textstyle\sum_{i\in I} d_\infty(x, x') && |x[i] - x'[i]| \leq d_\infty(x, x') \\
&= d_\infty(x, x')
\end{aligned}$$

For the p-perc query it holds that $p\text{-perc}(x) = \text{sort}(x)[l]$ and $p\text{-perc}(x') = \text{sort}(x')[l]$ for the same l (since $I_\varnothing(x) = I_\varnothing(x')$). Let $h, h' \in 1..n$ such that $x[h] = \text{sort}(x)[l]$ and $x'[h'] = \text{sort}(x')[l]$.

Assume that $x[h] \leq x[h']$ (the case $x[h] \geq x[h']$ is symmetric). By the definition of sort, there are at least l elements $j \in 1..n$ such that $x[j] \leq x[h]$ (including h itself). Moreover, there are at most $l - 1$ elements $j \in 1..n$ such that $x'[j] < x'[h']$. Hence, there exists at least one $j \in 1..n$ such that

$$x[j] \leq x[h] \qquad \text{and} \qquad x'[j] \geq x'[h']$$

It also holds that $|x[i] - x'[i]| \leq d_\infty(x, x')$, i.e.

$$x[i] - d_\infty(x, x') \leq x'[i] \leq x[i] + d_\infty(x, x') \quad \forall i \in 1..n \tag{4.3}$$

From $x[h] \leq x[h']$ and (4.3) we get

$$x[h] - d_\infty(x, x') \leq x'[h']$$

Moreover, it holds that

$$x'[h'] \leq x'[j] \leq x[j] + d_\infty(x, x') \leq x[h] + d_\infty(x, x')$$

thus

$$d_{\mathbb{R}}(p\text{-perc}(x), p\text{-perc}(x')) = |x[h] - x'[h']| \leq d_\infty(x, x')$$

For the "uniformly" part, let $y, y' \in q[0..k]$; we construct $x = \langle y, \varnothing, \ldots, \varnothing\rangle, x' = \langle y', \varnothing, \ldots, \varnothing\rangle$, for which it holds that $f(x) = y, f(x') = y'$ (for both queries) and $d_\infty(x, x') = d_{\mathbb{R}}(y, y')$.

Note that for p-perc we can construct x, x' without $\varnothing$ values with the same property. However, for the avg query this is not possible; its uniform optimality depends on the fact that $\varnothing$ values are allowed. If $\varnothing \notin \mathcal{V}$ then avg is essentially equivalent to sum, which is not uniformly optimal w.r.t. $d_\infty, d_{\mathbb{R}}$. □

Corollary. *TG_ϵ is f-ϵd_∞-optimal for $f \in \{\text{avg}, p\text{-perc}\}$, $\epsilon > 0$.*

4.5 Concluding remarks

Related Work

Several works in the differential privacy literature consider adjacency relations different than the standard one, effectively using a metric tailored to that application. Examples include group privacy [Dwork 2006a] and edge privacy for graphs [Nissim 2007].

The generalization of differential privacy to arbitrary metrics was considered also in [Barthe 2012, Reed 2010]. In those works, however, the purpose of extending the definition was to obtain compositional methods for proving differential privacy in programming languages, while in our work we focus on the implications of such extension for the theory of differential privacy. Namely, we aim at obtaining new meaningful definitions of privacy for various contexts through the use of different metrics (cf. the examples of the smart meters and of geolocation), and at investigating the existence of optimal mechanisms.

Another work closely related to ours is [Dwork 2012] in which an extended definition of differential privacy is used to capture the notion of fairness in classification. A metric d is used to model the fact that certain individuals are required to be classified similarly, and a mechanism satisfying d-privacy is considered fair, since it produces similar results for similar individuals. We view fairness as one of the many interesting notions that can be obtained through the use of metrics in various contexts, thus it encourages our goal of studying d-privacy. With respect to the actual metrics used in this chapter, the difference is that we consider metrics that depend on the individuals' values, while [Dwork 2012] considers metrics between individuals.

Summary

Starting from the observation that differential privacy requires that the distinguishability of two databases depends on their Hamming distance, we have explored the consequences of extending this principle to arbitrary metrics. In this way we have obtained a rich framework suitable to model a large variety of privacy problems, and in domains other than statistical databases. Furthermore, even in statistical databases applications, whenever the privacy concern is related to disclosing small variations in the values of the individuals (rather than large ones), then our framework allows a more precise calibration of the noise necessary for achieving the intended level of privacy, and this results, in general, in a better utility than the one achievable under the constraint of standard differential privacy. We have investigated the trade-off between privacy and utility in this extended setting, and it turns out changing the metric has considerable implications on the existence of universally optimal mechanisms. In particular, for the Manhattan distance, the normalized Manhattan distance, and the max distance it is possible to define universally optimal mechanisms for several common queries like the sum, the average, and the percentile. This contrast sharply with the case of standard differential privacy,

where universally optimal mechanisms exist only for counting queries. Moreover, in our framework it is possible to express queries that have unbounded sensitivity with respect to the standard Hamming metric, as long as this sensitivity is bounded with respect to another metric. Finally, we have shown the applicability of our framework to various privacy problems in a domains, from the usual case of statistical databases to smart meters.

Chapter 5

Privacy in Location Based Systems

In several situations it is desirable to know the location of an individual or a group of individuals in order to provide a service. For instance: In census-based statistics, to determine the population density in certain areas, in transportation industry, to estimate the average number of people who need to travel between two given stations, and in smartphone applications, to obtain points of interest nearby such as restaurants.

Due to privacy concerns, an individual may refuse to disclose his exact location to the service provider. Nevertheless, he may be willing to reveal approximate location information. It is worth noting that for several location-based systems it is usually enough to obtain an approximate location to be able to provide an accurate service. Note however, that in order to guarantee a non-negligible level of privacy, the random location cannot be generated naively. Therefore, if we want to develop a method to randomize location coordinates, we have to understand what kind of privacy the user expects to have, and how much information he is willing to reveal.

In this scenario, the privacy level depends on the accuracy with which an attacker can guess an individual's location from the reported one. We will therefore aim for a distance-dependent notion of privacy, requiring points that are close in distance to each other to be *indistinguishable* from the attacker's point of view. However, we still allow the service provider to distinguish between points that are far from each other. This is exactly the kind of situation in which the notion of d-privacy presented in Chapter 4 shows to be useful. In this particular case, the privacy guarantee can also be thought as an individual having a certain level of privacy *within a radius*. In this sense, we can say that the user enjoys a privacy level l within a radius r if any two locations at distance at most r produce observations with "similar" distributions, where the "level of similarity" depends on l. By considering the set of secrets $\mathcal{X}$ as the set containing all possible locations of an individual, we can see that this guarantee can be achieved by considering an instance of the more general notion of $d_{\mathcal{X}}$-privacy, taking $d_{\mathcal{X}} = \frac{l}{r} d_2$ as the privacy metric (recall that d_2 is the Euclidean distance). Moreover, it is clear that this instantiation provides a certain level of privacy for any radius: if $\epsilon = \frac{l}{r}$, then for a radius r' the privacy level is $l' = \epsilon r'$. We can therefore give a first, intuitive definition of our location privacy notion, that we call *geo-indistinguishability*:

> A location privacy mechanism satisfies ϵ-geo-indistinguishability if and only if for any radius $r > 0$, the user enjoys ϵr-privacy within r.

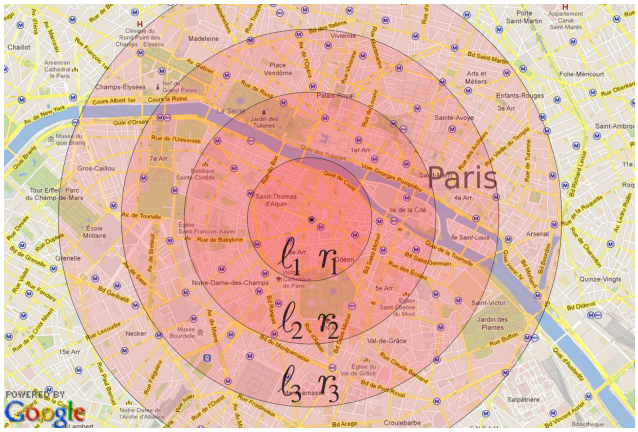


Figure 5.1: Geo-indistinguishability: privacy varying with r.

As stated before, this definition implies that the user is protected within any radius r, but with a level $\ell = \epsilon r$ that increases with the distance. Within a short radius, for instance $r = 1$ km, ℓ is small, guaranteeing that the provider cannot infer the user's location within, say, the 7th arrondissement of Paris. Farther away from the user, for instance for $r = 1000$ km, ℓ becomes large, allowing the location-based service (LBS) provider to infer that with high probability the user is located in Paris instead of, say, London. Figure 5.1 illustrates the idea of privacy levels decreasing with the radius.

In this chapter, we study the formal aspects of this privacy notion as an instance of the general definition of d-privacy presented in Chapter 4. First, we recall the different characterizations and properties of this notion, explaining what they mean in the context of location privacy. Secondly, we present a mechanism to achieve this privacy definition, briefly mentioned before in Section 4.2.1, and explain how to overcome two issues in the implementation, namely the truncation of the area of reported results and the discretization of the generated points. Thirdly, we present two case studies showing how to use the proposed method to enhance LBS applications with geo-indistinguishability guarantees, and how to sanitize a dataset containing geospatial information. Finally, we compare our mechanism with others in the literature using the privacy metric proposed in [Shokri 2012].

5.1 Geo-indistinguishability

In this section we formalize our notion of geo-indistinguishability, as an instance of Definition 4.1. In this particular case, the set of secrets $\mathcal{X}$ contains *points of interest*, typically the user's possible locations. The $\mathcal{Z}$ of reported values can in general be arbitrary, allowing to report obfuscated locations, cloaking regions, sets of locations, etc. However, to simplify the discussion, we sometimes consider $\mathcal{Z}$ to also contain spatial points, assuming an operational scenario of a user located at

$x \in \mathcal{X}$ and communicating to the adversary a randomly selected location $z \in \mathcal{Z}$ (e.g. an obfuscated point).

In this scenario, probabilities come into place in two ways. First, the attacker might have side information about the user's location, knowing, for example, that he is likely to be visiting the Eiffel Tower, while unlikely to be swimming in the Seine river. As discussed in previous chapters, the attacker's side information can be modeled by a *prior* distribution π on $\mathcal{X}$, where $\pi(x)$ is the probability assigned to the location x.

Second, the selection of a reported value $z \in \mathcal{Z}$ is itself probabilistic, since it is obtained by adding random noise to the actual location x, by using a probabilistic mechanism K; i.e. K is a function assigning to each location $x \in \mathcal{X}$ a probability distribution on $\mathcal{Z}$.

We can now state our definition of geo-indistinguishabilityas follows:

Definition 5.1 (GEO-INDISTINGUISHABILITY)**.** *A location privacy mechanism K satisfies ϵ-geo-indistinguishability if and only if for all $x, x' \in \mathcal{X}$:*

$$d_{\mathcal{P}}(K(x), K(x')) \leq \epsilon d_2(x, x')$$

Equivalently, the definition can be formulated as $K(x)(Z) \leq e^{\epsilon d_2(x,x')} K(x')(Z)$ for all $x, x' \in \mathcal{X}, Z \subseteq \mathcal{Z}$. Note that for all points x' within a radius r from x, the definition forces the corresponding distributions to be at most $l = \epsilon r$ distant.

A note on the unit of measurement

It is worth noting that, since ϵ corresponds to the privacy level for one unit of distance, it is affected by the unit in which distances are measured. For instance, assume that $\epsilon = 0.1$ and distances are measured in meters. The level of privacy for points one kilometer away is 1000ϵ, hence changing the unit to kilometers requires to set $\epsilon = 100$ in order for the definition to remain unaffected. In other words, if r is a physical quantity expressed in some unit of measurement, then ϵ has to be expressed in the inverse unit. In this thesis we omit the unit since the choice is orthogonal to our goals.

5.1.1 Characterizations

We will now recall the two operational characterizations of our generalized privacy notion presented in Section 4.1, which helps us provide intuitive interpretations of the privacy guarantees offered by geo-indistinguishability.

Adversary's conclusions under hiding

The first characterization uses the concept of a *hiding function* $\phi : \mathcal{X} \to \mathcal{X}$, which can be applied to the user's actual location before the mechanism K, so that the latter has only access to a hidden version $\phi(x)$ of the location, instead of the real location x. Intuitively, a location remains private if, regardless of his side knowledge

(captured by his prior distribution), an adversary draws the same conclusions (captured by his posterior distribution), regardless of whether hiding has been applied or not. However, if ϕ replaces locations in Paris with those in London, then clearly the adversary's conclusions will be greatly affected. Hence, we require that the effect on the conclusions depends on the maximum distance $d_2(\phi) = \sup_{x \in \mathcal{X}} d_2(x, \phi(x))$ between the real and hidden location.

Theorem 5.1. *A mechanism K satisfies ϵ-geo-indistinguishability iff for all $\phi : \mathcal{X} \to \mathcal{X}$, all priors π on $\mathcal{X}$, and all $Z \subseteq \mathcal{Z}$:*

$$d_{\mathcal{P}}(\sigma_1, \sigma_2) \leq 2\epsilon d_2(\phi) \qquad \textit{where} \qquad \sigma_1 = \mathbf{Bayes}(\pi, K, Z) \quad \sigma_2 = \mathbf{Bayes}(\pi, K \circ \phi, Z)$$

Recall that the above characterization compares two *posterior* distributions. Both σ_1, σ_2 can be substantially different than the initial knowledge π, in which case it means that an adversary does learn some information about the user's location.

Knowledge of an informed attacker

A different approach is to measure how much the adversary learns about the user's location, by comparing his prior and posterior distributions. However, since some information is allowed to be revealed by design, these distributions can be far apart. Still, we can consider an *informed* adversary who already knows that the user is located within a set $N \subseteq \mathcal{X}$. Let $d_2(N) = \sup_{x,x' \in N} d_2(x, x')$ be the maximum distance between points in x. Intuitively, the user's location remains private if, regardless of his prior knowledge within N, the knowledge obtained by such an informed adversary should be limited by a factor depending on $d_2(N)$. This means that if $d_2(N)$ is small, i.e. the adversary already knows the location with some accuracy, then the information that he obtains is also small, meaning that he cannot improve his accuracy. Denoting by $\pi_{|N}$ the distribution obtained from π by restricting to N (i.e. $\pi_{|N}(x) = \pi(x|N)$), we obtain the following characterization:

Theorem 5.2. *A mechanism K satisfies ϵ-geo-indistinguishability iff for all $N \subseteq \mathcal{X}$, all priors π on $\mathcal{X}$, and all $Z \subseteq \mathcal{Z}$:*

$$d_{\mathcal{P}}(\pi_{|N}, \sigma_{|N}) \leq \epsilon d_2(N) \qquad \textit{where} \qquad \sigma = \mathbf{Bayes}(\pi, K, Z)$$

Abstracting from side information

A major difference of geo-indistinguishability, compared to similar approaches from the literature, is that it abstracts from the side information available to the adversary, i.e. from the prior distribution. This is a subtle issue, and often a source of confusion, thus it is worth to clarify what "abstracting from the prior" means. The goal of a privacy definition is to restrict the information *leakage* caused by the observation. Note that the lack of leakage does not mean that the user's location

cannot be inferred (it could be inferred by the prior alone), but instead that the adversary's knowledge does not increase significantly *due to the observation*.

However, in the context of LBSs, no privacy definition can ensure a small leakage under any prior, and at the same time allow reasonable utility. Consider, for instance, an attacker who knows that the user is located at some airport, but not which one. The attacker's prior knowledge is very limited, still any useful LBS query should reveal at least the user's city, from which the exact location (i.e. the city's airport) can be inferred. Clearly, due to the side information, the leakage caused by the observation is high.

So, since we cannot eliminate leakage under any prior, how can we give a reasonable privacy definition without restricting to a particular one? First, we give a formulation (Definition 5.1) which does not involve the prior at all, allowing to verify it without knowing the prior. At the same time, we give two characterizations which explicitly quantify over all priors, shedding light on how the prior affects the privacy guarantees.

Finally, we should point out that differential privacy abstracts from the prior in exactly the same way. Contrary to what is sometimes believed, the user's value is *not protected* under any prior information. Recalling the well-known example from [Dwork 2006a], if the adversary knows that Terry Gross is two inches shorter than the average Lithuanian woman, then he can accurately infer the height, even if the average is released in a differentially private way (in fact no useful mechanism can prevent this leakage). Differential privacy does ensure that the risk is the same whether she participates in the database or not, but this might be misleading: it does not imply the lack of leakage, only that it will happen anyway, whether she participates or not!

5.1.2 Protecting location sets

So far, we have assumed that the user has a single location that he wishes to communicate to a service provider in a private way (typically his current location). In practice, however, it is common for a user to have multiple points of interest, for instance a set of past locations or a set of locations he frequently visits. In this case, the user might wish to communicate to the provider some information that depends on all points; this could be either the whole set of points itself, or some aggregate information, for instance their centroid. As in the case of a single location, privacy is still a requirement; the provider is allowed to obtain only approximate information about the locations, their exact value should be kept private. In this section, we discuss how ϵ-geo-indistinguishability extends to the case where the secret is a tuple of points $\mathbf{x} = (x_1, \ldots, x_n)$.

Similarly to the case of a single point, the notion of distance is crucial for our definition. We define the distance between two tuples of points $\mathbf{x} = (x_1, \ldots, x_n), \mathbf{x}' = (x'_1, \ldots, x'_n)$ as:

$$d_\infty(\mathbf{x}, \mathbf{x}') = \max_i d(x_i, x'_i)$$

Intuitively, the choice of metric follows the idea of reasoning within a radius r: when $d_\infty(\mathbf{x}, \mathbf{x}') \leq r$, it means that all x_i, x_i' are within distance r from each other. All definitions and results of this section can be then directly applied to the case of multiple points, by using d_∞ as the underlying metric. Enjoying a privacy level of ℓ within a radius r means that two tuples at most r away from each other, should produce distributions at most ℓ apart.

Reporting the whole set

A natural question then to ask is how we can obfuscate a tuple of points, by independently applying an existing mechanism K_0 to each individual point, and report the obfuscated tuple. Starting from a tuple $\mathbf{x} = (x_1, \ldots, x_n)$, we independently apply K_0 to each x_i obtaining a reported point z_i, and then report the tuple $\mathbf{z} = (z_1, \ldots, z_n)$. Thus, the probability that the combined mechanism K reports $\mathbf{z}$, starting from $\mathbf{x}$, is the product of the probabilities to obtain each point z_i, starting from the corresponding point x_i, i.e. $K(\mathbf{x})(\mathbf{z}) = \prod_i K_0(x_i)(z_i)$. [1]

The next question is what level of privacy does K satisfy. For simplicity, consider a tuple of only two points (x_1, x_2), and assume that K_0 satisfies ϵ-geo-indistinguishability. At first look, one might expect the combined mechanism K to also satisfy ϵ-geo-indistinguishability, however this is not the case. The problem is that the two points might be *correlated*, thus an observation about x_1 will reveal information about x_2 and vice versa. Consider, for instance, the extreme case in which $x_1 = x_2$. Having two observations about the same point reduces the level of privacy, thus we cannot expect the combined mechanism to provide the same level of privacy.

Still, if K_0 satisfies ϵ-geo-indistinguishability, then K can be shown to satisfy $n\epsilon$-geo-indistinguishability, i.e. a level of privacy that scales linearly with n. Due to this scalability issue, the technique of independently applying a mechanism to each point is only useful when the number of points is small. Still, this is sufficient for some applications, such as the case study of Section 5.3. Note, however, that this technique is by no means the best we can hope for: similarly to standard differential privacy [Blum 2008, Roth 2010], better results could be achieved by adding noise to the whole tuple $\mathbf{x}$, instead of each individual point.

Reporting an aggregate location

Another interesting case is when we need to report some aggregate information obtained by $\mathbf{x}$, for instance the centroid of the tuple. In general we might need to report the result of a query $f : \mathcal{X}^n \to \mathcal{X}$. Similarly to the case of standard differential privacy, we can compute the real answer $f(\mathbf{x})$ and the add noise by applying a mechanism K to it. If f is Δ-sensitive w.r.t. d, d_∞, meaning that $d(f(\mathbf{x}), f(\mathbf{x}')) \leq \Delta d_\infty(\mathbf{x}, \mathbf{x}')$ for all $\mathbf{x}, \mathbf{x}'$, and K satisfies geo-indistinguishability,

[1] For simplicity we consider probabilities of points here; a formal treatment of continuous mechanism would require to consider sets.

then the composed mechanism $K \circ f$ can be shown to satisfy $\Delta\epsilon$-geo-indistinguishability.

Note that when dealing with aggregate data, standard differential privacy becomes a viable option. However, one needs to also examine the loss of utility caused by the added noise. This highly depends on the application: differential privacy is suitable for publishing aggregate queries with *low sensitivity*, meaning that changes in a single individual have a relatively small effect on the outcome. On the other hand, location information often has high sensitivity. A trivial example is the case where we want to publish the complete tuple of points. But sensitivity can be high even for aggregate information: consider the case of publishing the centroid of 5 users located anywhere in the world. Modifying a single user can hugely affect their centroid, thus achieving differential privacy would require so much noise that the result would be useless. For geo-indistinguishability, on the other hand, one needs to consider the distance between points when computing the sensitivity. In the case of the centroid, a small (in terms of distance) change in the tuple has a small effect on the result, thus geo-indistinguishability can be achieved with much less noise.

5.2 The Planar Laplace Mechanism

In this section we present a method to generate noise so to satisfy geo-indistinguishability, based on an instance of the corresponding Laplace mechanism presented in Section 4.2.1. We model the location domain as a discrete[2] Cartesian plane with the standard notion of Euclidean distance. This model can be considered a good approximation of the Earth surface when the area of interest is not too large. In the rest of the section we develop our mechanism according to the following plan:

(a) First, we define a mechanism to achieve geo-indistinguishability in the ideal case of the continuous plane. For each actual location, this mechanism should generate a random point in a way that satisfies geo-indistinguishability on $\mathbb{R}^2$.

(b) Then, we discretize the mechanism by remapping each point generated according to (a) to the closest point in the discrete domain.

(c) Finally, we truncate the mechanism, so to report only points within the limits of the considered area.

5.2.1 A mechanism for the continuous plane

Following the above plan, we start by defining a mechanism for geo-indistinguishability on the continuous plane. The idea is that whenever the actual location is $x \in \mathbb{R}^2$, we report, instead, a point $z \in \mathbb{R}^2$ generated randomly according to the noise function. The latter needs to be such that the probabilities of reporting a point in a certain

[2]For applications with digital interface the domain of interest is discrete, since the representation of the coordinates of the points is necessarily finite.

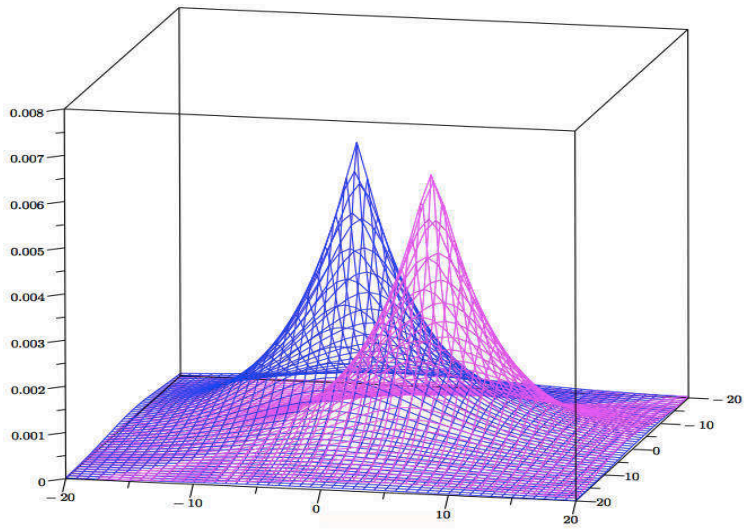


Figure 5.2: The pdf of two planar Laplacians, centered at $(-2, -4)$ and at $(5, 3)$ respectively, with $\epsilon = 1/5$.

(infinitesimal) area around z, when the actual locations are x and x' respectively, differs at most by a multiplicative factor $e^{-\epsilon\, d_2(x,x')}$.

We can achieve this property by using the Laplace mechanism for $\mathcal{Z} = \mathbb{R}^2$ presented in Section 4.2.1. This mechanism has the property that the probability of generating a point in the area around z decreases exponentially with the distance from the actual location x.

Given the parameter $\epsilon \in \mathbb{R}^+$, and the actual location $x \in \mathbb{R}^2$, the pdf of our noise mechanism, on any other point $z \in \mathbb{R}^2$, is:

$$D_\epsilon(x)(z) = \frac{\epsilon^2}{2\,\pi}\, e^{-\epsilon\, d_2(x,z)} \tag{5.1}$$

where $\epsilon^2/2\,\pi$ is a normalization factor. We call this function *planar Laplacian centered at* x. The corresponding distribution is illustrated in Figure 5.2. It is possible to show that (i) the projection of a planar Laplacian on any vertical plane passing by the center gives a (scaled) linear Laplacian, and (ii) the corresponding mechanism satisfies ϵ-geo-indistinguishability.

Drawing a random point

We illustrate now how to draw a random point from the pdf defined in (5.1). First of all, we note that the pdf of the planar Laplacian depends only on the distance from x. It will be convenient, therefore, to switch to a system of polar coordinates with origin in x. A point z will be represented as a point (r, θ), where r is the distance of z from x, and θ is the angle that the line $z\,x$ forms with respect to the horizontal axis of the Cartesian system. Following the standard transformation formula, the

pdf of the *polar Laplacian* centered at the origin (x) is:

$$D_\epsilon(r,\theta) = \frac{\epsilon^2}{2\,\pi}\, r\, e^{-\epsilon\, r} \tag{5.2}$$

We note now that the polar Laplacian defined above enjoys a property that is very convenient for drawing in an efficient way: *the two random variables that represent the radius and the angle are independent.* Namely, the pdf can be expressed as the product of the two marginals. In fact, let us denote these two random variables by R (the radius) and Θ (the angle). The two marginals are:

$$D_{\epsilon,R}(r) \;=\; \int_0^{2\pi} D_\epsilon(r,\theta)\, d\theta \;=\; \epsilon^2\, r\, e^{-\epsilon\, r}$$

$$D_{\epsilon,\Theta}(\theta) \;=\; \int_0^{\infty} D_\epsilon(r,\theta)\, dr \;=\; \tfrac{1}{2\,\pi}$$

Hence we have $D_\epsilon(r,\theta) = D_{\epsilon,R}(r)\; D_{\epsilon,\Theta}(\theta)$. Note that $D_{\epsilon,R}(r)$ corresponds to the pdf of the *gamma distribution* with shape 2 and scale $1/\epsilon$.

Thanks to the fact that R and Θ are independent, in order to draw a point (r,θ) from $D_\epsilon(r,\theta)$ it is sufficient to draw separately r and θ from $D_{\epsilon,R}(r)$ and $D_{\epsilon,\Theta}(\theta)$ respectively.

Since $D_{\epsilon,\Theta}(\theta)$ is constant, drawing θ is easy: it is sufficient to generate θ as a random number in the interval $[0, 2\pi)$ with uniform distribution.

We now show how to draw r. Following standard lines, we consider the cumulative distribution function (cdf) $C_\epsilon(r)$:

$$C_\epsilon(r) \;=\; \int_0^r D_{\epsilon,R}(\rho) d\rho \;=\; 1 - (1+\epsilon\, r)\, e^{-\epsilon\, r}$$

Intuitively, $C_\epsilon(r)$ represents the probability that the radius of the random point falls between 0 and r. Finally, we generate a random number p with uniform probability in the interval $[0, 1)$, and we set $r = C_\epsilon^{-1}(p)$. Note that

$$C_\epsilon^{-1}(p) = -\tfrac{1}{\epsilon}\big(W_{-1}(\tfrac{p-1}{e}) + 1\big)$$

where W_{-1} is the Lambert W function (the -1 branch), which can be computed efficiently and is implemented in several numerical libraries (MATLAB, Maple, GSL, ...).

5.2.2 Discretization

We discuss now how to approximate the Laplace mechanism on a grid $\mathcal{G}$ of discrete Cartesian coordinates. Let us recall the points (a) and (b) of the plan, in light of the development so far: Given the actual location x_0, report the point x in $\mathcal{G}$ obtained as follows:

(a) first, draw a point (r,θ) following the method in Figure 5.3,

(b) then, remap (r,θ) to the closest point x on $\mathcal{G}$.

Drawing a point (r, θ) from the polar Laplacian
1. draw θ uniformly in $[0, 2\pi)$
2. draw p uniformly in $[0, 1)$ and set $r = C_\epsilon^{-1}(p)$

Figure 5.3: Method to generate Laplacian noise.

We will denote by $K_\epsilon : \mathcal{G} \to \mathcal{P}(\mathcal{G})$ the above mechanism. In summary, $K_\epsilon(x)(z)$ represents the probability of reporting the point z when the actual point is x.

It is not obvious that the discretization preserves geo-indistinguishability, due to the following problem: In principle, each point x in $\mathcal{G}$ should gather the probability of the set of points for which x is the closest point in $\mathcal{G}$, namely

$$R(x) = \{y \in \mathbb{R}^2 \,|\, \forall x' \in \mathcal{G}.\ d(x, y) \leq d(x', y)\}$$

However, due to the finite precision of the machine, the noise generated according to (a) is already discretized in accordance with the polar system. Let $\mathcal{W}$ denote the discrete set of points actually generated in (a). Each of those points (r, θ) is drawn with the probability of the area between r, $r + \delta_r$, θ and $\theta + \delta_\theta$, where δ_r and δ_θ denote the precision of the machine in representing the radius and the angle respectively. Hence, step (b) generates a point x in $\mathcal{G}$ with the probability of the set $R_\mathcal{W}(x) = R(x) \cap \mathcal{W}$. This introduces some irregularity in the mechanism, because the region associated to $R_\mathcal{W}(x)$ has a different shape and area depending on the position of x relatively to x_0. The situation is illustrated in Figure 5.4 with $R_0 = R_\mathcal{W}(x_0)$ and $R_1 = R_\mathcal{W}(x_1)$.

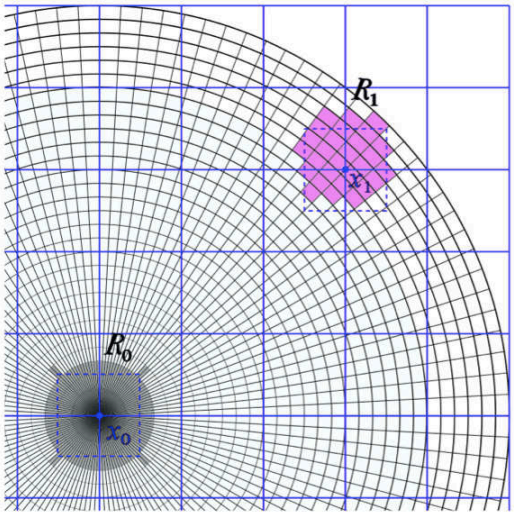


Figure 5.4: Remapping the points in polar coordinates to points in the grid.

Geo-indistinguishability of the discretized mechanism

We now analyze the privacy guarantees provided by our discretized mechanism. We show that the discretization preserves geo-indistinguishability, at the price of a degradation of the privacy parameter ϵ.

For the sake of generality we do not require the step units along the two dimensions of $\mathcal{G}$ to be equal. We will call them *grid units*, and will denote by u and v the smaller and the larger unit, respectively. We recall that δ_θ and δ_r denote the precision of the machine in representing θ and r, respectively. We assume that $\delta_r \leq r_{\max}\delta_\theta$. The following theorem states the geo-indistinguishability guarantees provided by our mechanism: $K_{\epsilon'}$ satisfies ϵ-geo-indistinguishability, within a range $r_{\max}$, provided that ϵ' is chosen in a suitable way that depends on ϵ, on the length of the step units of $\mathcal{G}$, and on the precision of the machine.

Theorem 5.3. *Assume $r_{\max} < u/\delta_\theta$, and let $q = u/r_{\max}\delta_\theta$. Let $\epsilon, \epsilon' \in \mathbb{R}^+$ such that*

$$\epsilon' + \frac{1}{u} \ln \frac{q + 2\, e^{\epsilon' u}}{q - 2\, e^{\epsilon' u}} \leq \epsilon$$

Then $K_{\epsilon'}$ provides ϵ-geo-indistinguishability within the range of $r_{\max}$. Namely, if $d(x,z), d(x',z) \leq r_{\max}$ then:

$$K_{\epsilon'}(x)(z) \leq e^{\epsilon\, d(x,x')} K_{\epsilon'}(x')(z).$$

Proof. The case in which $x = x'$ is trivial. We consider therefore only the case in which $x \neq x'$. Note that in this case $d(x,x') \geq u$. We proceed by determining an upper bound on $K_{\epsilon'}(x)(z)$ and a lower bound on $K_{\epsilon'}(x')(z)$ for generic x, x' and z such that $d(x,z), d(x',z) \leq r_{\max}$. Let S be the set of points for which z is the closest point in $\mathcal{G}$, namely:

$$S = R(z) = \{y \in \mathbb{R}^2 \mid \forall z' \in \mathcal{G}.\ d(y,z) \leq d(y,z')\}$$

Ideally, the points remapped in z would be exactly those in S. However, as discussed before, the points actually remapped in z are those of $R_{\mathcal{W}}(z)$ Hence the probability of z is that of S plus or minus the small rectangles[3] W of size $\delta_r \times r\, \delta_\theta$ at the border of S, where $r = d(x,z)$, see Figure 5.5. Let us denote by S_W the total area of these small rectangles W on one of the sides of S. Since $d(x,z) \leq r_{\max} < u/\delta_\theta$, and $\delta_r < r_{\max}\delta_\theta$, we have that S_W is less than $1/q$ of the area of S, where $q = u/r_{\max}\delta_\theta$. The probability density on this area differs at most by a factor $e^{\epsilon' u}$ from that of the other points in S. Finally, note that on two sides of S the rectangles W contribute positively to $K_{\epsilon'}(x)(z)$, while on two sides they contribute negatively. Summarizing, we have:

$$K_{\epsilon'}(x)(z) \ \leq\ (1 + \frac{2\, e^{\epsilon' u}}{q}) \int_S D_{\epsilon'}(x)(s)ds \tag{5.3}$$

[3] W is actually a fragment of a circular crown, but since δ_θ is very small, it approximates a rectangle. Also, the side of W is not exactly $r\,\delta_\theta$, it is a number in the interval $[(r - u/\sqrt{2})\,\delta_\theta, (r + u/\sqrt{2})\,\delta_\theta]$. However $u/\sqrt{2}\,\delta_\theta$ is very small with respect to the other quantities involved, hence we consider negligible this difference.

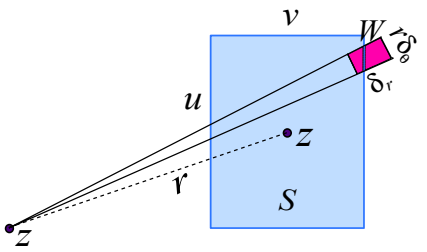


Figure 5.5: Bounding the probability of z in the discrete Laplacian.

and

$$(1 - \frac{2\, e^{\epsilon' u}}{q}) \int_S D_{\epsilon'}(x')(s)ds \;\leq\; K_{\epsilon'}(x')(z) \tag{5.4}$$

Observe now that

$$\frac{D_{\epsilon'}(x)(s)}{D_{\epsilon'}(x')(s)} = e^{-\epsilon'(d(x,s)-d(x',s))}$$

By triangular inequality we obtain

$$D_{\epsilon'}(x)(s) \leq e^{\epsilon'\, d(x,x')} D_{\epsilon'}(x')(s)$$

from which we derive

$$\int_S D_{\epsilon'}(x)(s)ds \leq e^{\epsilon'\, d(x,x')} \int_S D_{\epsilon'}(x')(s)ds \tag{5.5}$$

from which, using (5.3), (5.5), and (5.4), we obtain

$$K_{\epsilon'}(x)(z) \leq e^{\epsilon'\, d(x,x')}\, K_{\epsilon'}(x')(z)\, \frac{q + 2\, e^{\epsilon' u}}{q - 2\, e^{\epsilon' u}} \tag{5.6}$$

Assume now that

$$\epsilon' + \frac{1}{u} \ln \frac{q + 2\, e^{\epsilon' u}}{q - 2\, e^{\epsilon' u}} \leq \epsilon$$

Since we are assuming $d(x, x') \geq u$, we derive:

$$e^{\epsilon'\, d(x,x')}\, \frac{q + 2\, e^{\epsilon' u}}{q - 2\, e^{\epsilon' u}} \leq e^{\epsilon\, d(x,x')} \tag{5.7}$$

Finally, from (5.6) and (5.7), we conclude. □

The difference between ϵ' and ϵ represents the additional noise needed to compensate the effect of discretization. Note that $r_{\max}$, which determines the area in which ϵ-geo-indistinguishability is guaranteed, must be chosen in such a way that $q > 2\, e^{\epsilon' u}$. Furthermore there is a trade-off between ϵ' and $r_{\max}$: If we want ϵ' to be close to ϵ then we need q to be large. Depending on the precision, this may or may not imply a serious limit on $r_{\max}$. Vice versa, if we want $r_{\max}$ to be large then, depending on the precision, ϵ' may need to be significantly smaller than ϵ, and furthermore we may have a constraint on the minimum possible value for ϵ,

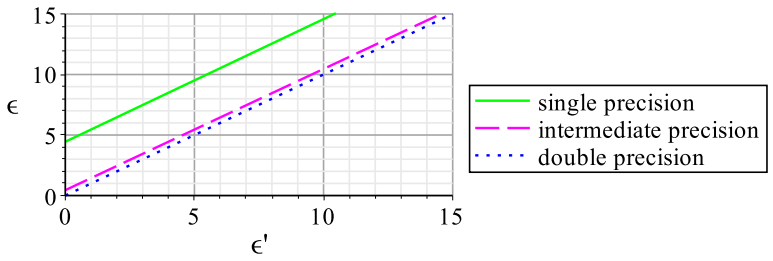


Figure 5.6: The relation between ϵ and ϵ' for $r_{\max} = 10^2$ km.

which means that we may not have the possibility of achieving an arbitrary level of geo-indistinguishability.

Figure 5.6 shows how the additional noise varies depending on the precision of the machine. In this figure, $r_{\max}$ is set to be 10^2 km, and we consider the cases of double precision (16 significant digits, i.e., $\delta_\theta = 10^{-16}$), single precision (7 significant digits), and an intermediate precision of 9 significant digits. Note that with double precision the additional noise is negligible.

Note that in Theorem 5.3 the restriction about $r_{\max}$ is crucial. Namely, ϵ-geo-indistinguishability does not hold for arbitrary distances for any finite ϵ. Intuitively, this is because the step units of $\mathcal{W}$ (see Figure 5.4) become larger with the distance r from x_0. The step units of $\mathcal{G}$, on the other hand, remain the same. When the steps in $\mathcal{W}$ become larger than those of $\mathcal{G}$, some x's have an empty $R_{\mathcal{W}}(x)$. Therefore when x is far away from x_0 its probability may or may not be 0, depending on the position of x_0 in $\mathcal{G}$, which means that geo-indistinguishability cannot be satisfied.

5.2.3 Truncation

The Laplace mechanisms described in the previous sections have the potential to generate points everywhere in the plane, which causes several issues: first, digital applications have finite memory, hence these mechanisms are not implementable. Second, the discretized mechanism of Section 5.2.2 satisfies geo-indistinguishability only within a certain range, not on the full plane. Finally, in practical applications we are anyway interested in locations within a finite region (the earth itself is finite), hence it is desirable that the reported location lies within that region. For the above reasons, we propose a truncated variant of the discretized mechanism which generates points only within a specified region and fully satisfies geo-indistinguishability. The full mechanism (with discretization and truncation) is referred to as "Planar Laplace mechanism" and denoted by PL_ϵ.

We assume a finite set $\mathcal{A} \subset \mathbb{R}^2$ of admissible locations, with diameter $\mathrm{diam}(\mathcal{A})$ (maximum distance between points in $\mathcal{A}$). This set is *fixed*, i.e. it does not depend on the actual location x_0. Our truncated mechanism $PL_\epsilon : \mathcal{A} \to \mathcal{P}(\mathcal{A} \cap \mathcal{G})$ works like the discretized Laplacian of the previous section, with the difference that the point generated in step (a) is remapped to the closest point in $\mathcal{A} \cap \mathcal{G}$. The complete

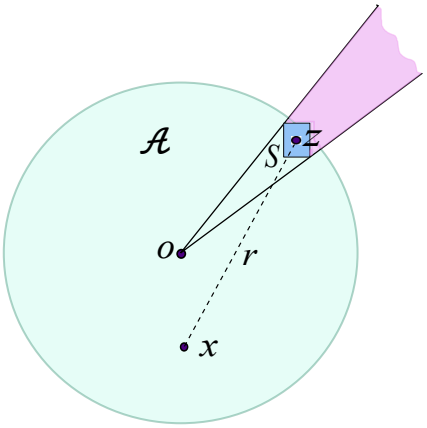


Figure 5.7: Probability of z in the truncated discrete laplacian.

mechanism is shown in Figure 5.8; note that step 1 assumes that $\text{diam}(\mathcal{A}) < u/\delta_\theta$, otherwise no such ϵ' exists.

Theorem 5.4. *PL_ϵ satisfies ϵ-geo-indistinguishability. namely*

$$K^T_{\epsilon'}(x)(z) \leq e^{\epsilon\, d(x,x')} K^T_{\epsilon'}(x')(z) \quad \textit{for every } x, x' \in \mathcal{A}$$

Proof. The proof proceeds like the one for Theorem 5.3, except when $R(z)$ is on the border of $\mathcal{A}$. In this latter case, the probability on z is given not only by the probability on $R(z)$ (plus or minus the small rectangles W – see the proof of Theorem 5.3), but also by the probability of the part C of the cone determined by o, $R(z)$, and lying outside $\mathcal{A}$ (see Figure 5.7). Following a similar reasoning as in the proof of Theorem 5.3 we get

$$K^T_{\epsilon'}(x)(z) \;\leq\; (1 + \frac{2\, e^{\epsilon' u}}{q}) \int_{S \cup C} D_{\epsilon'}(x)(s) ds$$

and

$$(1 - \frac{2\, e^{\epsilon' u}}{q}) \int_{S \cup C} D_{\epsilon'}(x')(s) ds \;\leq\; K^T_{\epsilon'}(x')(z)$$

The rest follows as in the proof of Theorem 5.3. □

5.3 Enhancing LBSs with Privacy

In this section we present a case study of our privacy mechanism in the context of LBSs. In particular we show how to enhance LBS applications with privacy guarantees while still providing a high quality service to their users. We assume a simple client-server architecture where users communicate via a trusted mobile application (the client – typically installed in a smart-phone) with an unknown/untrusted LBS provider (the server – typically running on the cloud). Hence, in contrast to

Input: x *// point to sanitize*

ϵ *// privacy parameter*

u, v, δ_θ, δ_r *// precision parameters*

$\mathcal{A}$ *// acceptable locations*

Output: Sanitized version z of input x

1. $\epsilon' \leftarrow \max \epsilon'$ satisfying Thm 5.3 for $r_{\max} = \mathrm{diam}(\mathcal{A})$
2. draw θ unif. in $[0, 2\pi)$ *// draw angle*
3. draw p unif. in $[0, 1)$, set $r \leftarrow C_{\epsilon'}^{-1}(p)$ *// draw radius*
4. $z \leftarrow x + \langle r\cos(\theta), r\sin(\theta)\rangle$ *// to cartesian, add vectors*
5. $z \leftarrow \mathrm{closest}(z, \mathcal{A})$ *// truncation*
6. **return** z

Figure 5.8: The Planar Laplace mechanism PL_ϵ

other solutions proposed in the literature, our approach does not rely on trusted third-party servers.

In the following we distinguish between *mildly-location-sensitive* and *highly-location-sensitive* LBS applications. The former category corresponds to LBS applications offering a service that does not heavily rely on the precision of the location information provided by the user. Examples of such applications are weather forecast applications and LBS applications for retrieval of certain kind of points of interest (POI), like gas stations or cinemas. Enhancing this kind of LBSs with geo-indistinguishability is relatively straightforward as it only requires to obfuscate the user's location using the Planar Laplace mechanism (Figure 5.8).

Our running example lies within the second category: For the user sitting at Café Les Deux Magots, information about restaurants nearby Champs Élysées is considerably less valuable than information about restaurants around his location. Enhancing highly-location-sensitive LBSs with privacy guarantees is more challenging. Our approach consists on implementing the following three steps:

1. Implement the Planar Laplace mechanism (Figure 5.8) on the client application in order to report to the LBS server the user's obfuscated location z rather than his real location x.

2. Due to the fact that the information retrieved from the server is about POI nearby z, the area of POI information retrieval should be increased. In this way, if the user wishes to obtain information about POI within, say, 300 m of x, the client application should request information about POI within, say, 1 km of z. Figure 5.9 illustrates this situation. We will refer to the blue circle as *area of interest* (AOI) and to the grey circle as *area of retrieval* (AOR).

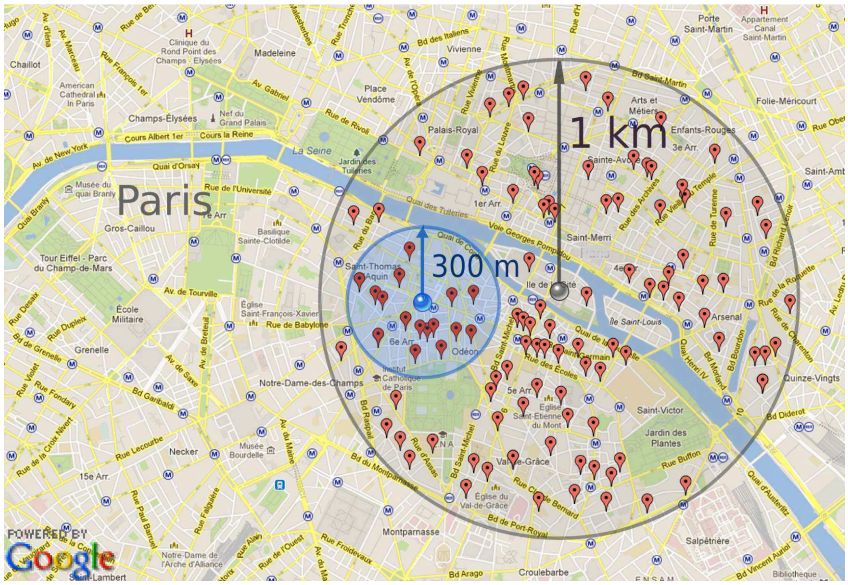


Figure 5.9: AOI and AOR of 300 m and 1 km radius respectively.

3. Finally, the client application should filter the retrieved POI information (depicted by the pins within the area of retrieval in Figure 5.9) in order to provide to the user with the desired information (depicted by pins within the user's area of interest in Figure 5.9).

Ideally, the AOI should always be fully contained in the AOR. Unfortunately, due to the probabilistic nature of our perturbation mechanism, this condition cannot be guaranteed (note that the AOR is centered on a randomly generated location that can be arbitrarily distant from the real location). It is also worth noting that the client application cannot dynamically adjust the radius of the AOR in order to ensure that it always contains the AOI as this approach would completely jeopardize the privacy guarantees: on the one hand, the size of the AOR would leak information about the user's real location and, on the other hand, the LBS provider would know with certainty that the user is located within the retrieval area. Thus, in order to provide geo-indistinguishability, the AOR has to be defined *independently* from the randomly generated location.

Since we cannot guarantee that the AOI is fully contained in the AOR, we introduce the notion of *accuracy*, which measures the probability of such event. In the following, we will refer to an LBS application in abstract terms, as characterized by a location perturbation mechanism K and a fixed AOR radius. We use rad_R and rad_I to denote the radius of the AOR and the AOI, respectively, and $\mathcal{B}(x, r)$ to denote the circle with center x and radius r.

5.3.1 On the accuracy of LBSs

Intuitively, an LBS application is (c, rad_I)-accurate if the probability of the AOI to be fully contained in the AOR is bounded from below by a *confidence factor* c. Formally:

Definition 5.2 (LBS application accuracy)**.** *An LBS application (K, rad_R) is (c, rad_I)-accurate iff for all locations x we have that $\mathcal{B}(x, rad_I)$ is fully contained in $\mathcal{B}(K(x), rad_R)$ with probability at least c.*

Given a privacy parameter ϵ and accuracy parameters (c, rad_I), our goal is to obtain an LBS application (K, rad_R) satisfying both ϵ-geo-indistinguishability and (c, rad_I)-accuracy. As a perturbation mechanism, we use the Planar Laplace PL_ϵ (Figure 5.8), which satisfies ϵ-geo-indistinguishability. As for rad_R, we aim at finding the minimum value validating the accuracy condition. Finding such minimum value is crucial to minimize the bandwidth overhead inherent to our proposal. In the following we will investigate how to achieve this goal by *statically* defining rad_R as a function of the mechanism and the accuracy parameters c and rad_I.

For our purpose, it will be convenient to use the notion of (α, δ)-usefulness, which was introduced in [Blum 2008]. A location perturbation mechanism K is (α, δ)-*useful* if for every location x the reported location $z = K(x)$ satisfies $d(x, z) \leq \alpha$ with probability at least δ. In the case of the Planar Laplace, it is easy to see that, by definition, the α and δ values which express its usefulness are related by C_ϵ [4], the cdf of the Gamma distribution:

Observation 5.1. *For any $\alpha > 0$, PL_ϵ is (α, δ)-useful if $\alpha \leq C_\epsilon^{-1}(\delta)$.*

Figure 5.10 illustrates the (α, δ)-usefulness of PL_ϵ for $r = 0.2$ (as in our running example) and various values of ℓ (recall that $\ell = \epsilon\, r$). It follows from the figure that a mechanism providing the privacy guarantees specified in our running example (ϵ-geo-indistinguishability, with $\ell = \ln(4)$ and $r = 0.2$) generates an approximate location z falling within 1 km of the user's location x with probability 0.992, falling within 690 meters with probability 0.95, falling within 560 meters with probability 0.9, and falling within 390 meters with probability 0.75.

We now have all the necessary ingredients to determine the desired rad_R: By definition of usefulness, if PL_ϵ is (α, δ)-useful then the LBS application (PL_ϵ, rad_R) is (δ, rad_I)-accurate if $\alpha \leq rad_R - rad_I$. The converse also holds if δ is maximal. By Observation 5.1, we then have:

Proposition 5.1. *The LBS application (PL_ϵ, rad_R) is (c, rad_I)-accurate if $rad_R \geq rad_I + C_\epsilon^{-1}(c)$.*

Therefore, it is sufficient to set $rad_R = rad_I + C_\epsilon^{-1}(c)$.

[4] For simplicity we assume that $\epsilon' = \epsilon$ (see Figure 5.8), since their difference is negligible under double precision.

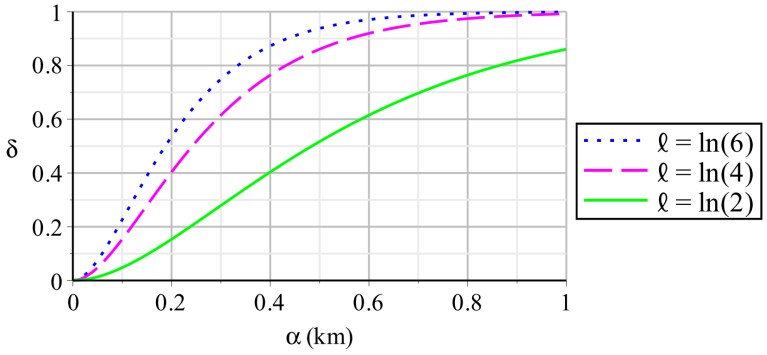


Figure 5.10: (α, δ)**-usefulness for** $r = 0.2$ **and various values of** ℓ**.**

Coming back to our running example ($\epsilon = \ln(4)/0.2$ and $rad_I = 0.3$), taking a confidence factor c of, say, 0.95, leads to a $(0.69, 0.95)$-useful mechanism (because $C_\epsilon^{-1}(c) = 0.69$). Thus, $(PL_\epsilon, 0.99)$ is both $^{\ln(4)}/_{0.2}$-geo-indistinguishable and $(0.95, 0.3)$-accurate.

5.3.2 Bandwidth overhead analysis

As expressed by Proposition 5.1, in order to implement an LBS application enhanced with geo-indistinguishability and accuracy it suffices to use the Planar Laplace mechanism and retrieve POIs for an enlarged radius rad_R. For each query made from a location x, the application needs to (i) obtain $z = PL_\epsilon(x)$, (ii) retrieve POIs for $\text{AOR} = \mathcal{B}(z, rad_R)$, and (iii) filter the results from AOR to AOI (as explained in step 3 above). Such implementation is straightforward and computationally efficient for modern smart-phone devices. In addition, it provides great flexibility to application developer and/or users to specify their desired/allowed level of privacy and accuracy. This, however, comes at a cost: bandwidth overhead.

In the following we turn our attention to investigating the bandwidth overhead yielded by our approach. We will do so in two steps: first we investigate how the AOR size increases for different privacy and LBS-specific parameters, and then we investigate how such increase translates into bandwidth overhead.

Figure 5.11 depicts the overhead of the AOR versus the AOI (represented as their ratio) when varying the level of confidence (c) and privacy (ℓ) and for fixed values $rad_I = 0.3$ and $r = 0.2$. The overhead increases slowly for levels of confidence up to 0.95 (regardless of the level of privacy) and increases sharply thereafter, yielding to a worst case scenario of a about 50 times increase for the combination of highest privacy ($\ell = \log(2)$) and highest confidence ($c = 0.99$).

In order to understand how the AOR increase translates into bandwidth overhead, we now investigate the density (in km^2) and size (in KB) of POIs by means of the Google Places API [Google Places]. This API allows to retrieve POIs' information for a specific `location`, `radius` around the location, and POI's `type` (among many other optional parameters). For instance, the HTTPS request:

```
https://maps.googleapis.com/maps/api/place/nearbysearch/json?location=
```

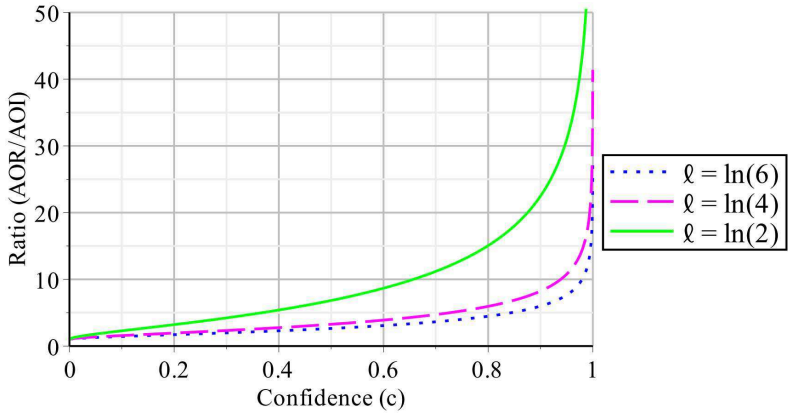


Figure 5.11: AOR vs AOI ratio for various levels of privacy and accuracy (using fixed $r = 0.2$ and $rad_I = 0.3$).

```
48.85412,2.33316&radius=300&types=restaurant&key=myKey
```

returns information (in JSON format) including location, address, name, rating, and opening times for all restaurants up to 300 meters from the location (48.85412, 2.33316) – which corresponds to the coordinates of Café Les Deux Magots in Paris.

We have used the APIs `nearbysearch` and `radarsearch` to calculate the average number of POIs per km^2 and the average size of POIs' information (in KB) respectively. We have considered two queries: restaurants in Paris, and restaurants in Buenos Aires. Our results show that there is an average of 137 restaurants per km^2 in Paris and 22 in Buenos Aires, while the average size per POI is 0.84 KB.

Combining this information with the AOR overhead depicted in Figure 5.11, we can derive the average bandwidth overhead for each query and various combinations of privacy and accuracy levels. For example, using the parameter combination of our running example (privacy level $\epsilon = \log(4)/0.2$, and accuracy level $c = 0.95, rad_I = 0.3$) we have a 10.7 ratio for an average of 38 ($\backsimeq (137/1000^2) \times (300^2 \times \pi)$) restaurants in the AOI. Thus the estimated bandwidth overhead is $39 \times (10.7 - 1) \times 0.84$KB $\backsimeq$ 318 KB.

Table 5.1 shows the bandwidth overhead for restaurants in Paris and Buenos Aires for the various combinations of privacy and accuracy levels. Looking at the worst case scenario, from a bandwidth overhead perspective, our combination of highest levels of privacy and accuracy (taking $\ell = \log(2)$ and $c = 0.99$) with the query for restaurants in Paris (which yields to a large number of POIs – significantly larger than average) results in a significant bandwidth overhead (up to 1.7MB). Such overhead reduces sharply when decreasing the level of privacy (e.g., from 1.7 MB to 557 KB when using $\ell = \log(4)$ instead of $\ell = \log(2)$). For more standard queries yielding a lower number of POIs, in contrast, even the combination of highest privacy and accuracy levels results in a relatively insignificant bandwidth overhead.

Concluding our bandwidth overhead analysis, we believe that the overhead necessary to enhance an LBS application with geo-indistinguishability guarantees is

Restaurants in Paris		**Accuracy** $rad_I = 0.3$		
		$c = 0.9$	$c = 0.95$	$c = 0.99$
Privacy $r = 0.2$	$\ell = \log(6)$	162 KB	216 KB	359 KB
	$\ell = \log(4)$	235 KB	318 KB	539 KB
	$\ell = \log(2)$	698 KB	974 KB	1.7 MB

Restaurants in Buenos Aires		**Accuracy** $rad_I = 0.3$		
		$c = 0.9$	$c = 0.95$	$c = 0.99$
Privacy $r = 0.2$	$\ell = \log(6)$	26 KB	34 KB	54 KB
	$\ell = \log(4)$	38 KB	51 KB	86 KB
	$\ell = \log(2)$	112 KB	156 KB	279 KB

Table 5.1: Bandwidth overhead for restaurants in Paris and in Buenos Aires for various levels of privacy and accuracy.

not prohibitive even for scenarios resulting in high bandwidth overhead (i.e., when combining very high privacy and accuracy levels with queries yielding a large number of POIs). Note that 1.7MB is comparable to 35 seconds of Youtube streaming or 80 seconds of standard Facebook usage [Vodafone]. Nevertheless, for cases in which minimizing bandwidth consumption is paramount, we believe that trading bandwidth consumption for privacy (e.g., using $\ell = \log(4)$ or even $\ell = \log(6)$) is an acceptable solution.

5.3.3 Further challenges: using an LBS multiple times

As discussed in Section 5.1.2, geo-indistinguishability can be naturally extended to multiple locations. In short, the idea of enjoying a *privacy level ℓ within r* remains the same but for all locations simultaneously. In this way the locations, say, x_1, x_2 of a user employing the LBS twice remain indistinguishable from all pair of locations at (point-wise) distance at most r (i.e., from all pairs x_1', x_2' such that $d(x_1, x_1') \leq r$ and $d(x_2, x_2') \leq r$).

A simple way of obtaining geo-indistinguishability guarantees when performing multiple queries is to employ our technique for protecting single locations to *independently* generate approximate locations for each of the user's locations. In this way, a user performing n queries via a mechanism providing ϵ-geo-indistinguishability enjoys $n\epsilon$-geo-indistinguishability (see Section 5.1.2).

This solution might be satisfactory when the number of queries to perform remains fairly low, but in other cases impractical, due to the privacy degradation. It is worth noting that the canonical technique for achieving standard differential privacy (based on adding noise according to the Laplace distribution) suffers of the same privacy degradation problem (ϵ increases linearly on the number of queries). Several articles in the literature focus on this problem (see [Roth 2010] for instance). We believe that the principles and techniques used to deal with this problem for standard differential privacy could be adapted to our scenario (either directly or motivationally).

5.4 Sanitizing datasets: US census case study

In this section we present a sanitation algorithm for datasets containing geographical information. We consider a realistic case study involving publicly available data developed by the U.S Census Bureau's Longitudinal Employer-Household Dynamics Program (LEHD). These data, called LEHD Origin-Destination Employment Statistics (LODES), are used by OnTheMap, a web-based interactive application developed by the US Census Bureau. The application enables, among other features, visualization of geographical information involving the residence and working location of US residents (e.g., distance from home to work location).

The LODES dataset includes information of the form $(hBlock, wBlock)$, where each pair represents a worker, the attribute $hBlock$ is the census block in which the worker lives, and $wBlock$ is the census block where the worker works. From this dataset it is possible to derive, by mapping home and work census blocks into their corresponding geographic centroids, a dataset with geographic information of the form ($hCoord$, $wCoord$), where each of the coordinate pairs corresponds to a census block pair.

The Census Bureau uses a *synthetic data generation algorithm* [Rubin 1993, Machanavajjhala 2008] to sanitize the LODES dataset. Roughly speaking, the algorithm interprets the dataset as an histogram where each $(hBlock, wBlock)$ pair is represented by a histogram bucket, the synthetic data generation algorithm sanitizes data by modifying the counts of the histogram.

In the following we present a sanitizing algorithm for datasets with geographical information (e.g. the LODES dataset) that provides geo-indistinguishability guarantees under the assumption that the home census blocks values in the dataset are uncorrelated. Although this assumption weakens the privacy guarantees provided by geo-indistinguishability, we believe that due to the anonymizing techniques applied by the Census Bureau to the released data involving census participants' information and to the large number of ($hCoord$, $wCoord$) pairs within small areas contained in the dataset, a practical attack based on correlation of points is unlikely.

Our sanitizing algorithm, described in Figure 5.12, takes as input (1) a dataset D to sanitize, (2) the privacy parameters ℓ and r (see Section 5.1), and (3) the precision parameters u, v, δ_r and δ_θ, and the region $\mathcal{A}$. (see Section 5.2.2) and

Sanitizing Algorithm for a Dataset of Locations

Input: $D : hCoord \times wCoord$ *// dataset to sanitize*

ℓ, r, u, v, δ_r, δ_θ, $\mathcal{A}$ *// same as in Figure 5.8*

Output: Sanitized version D' of input D

1. $D' = \emptyset$; *// initializing output dataset*
2. $\epsilon = \ell/r$;
3. **for each** $(c_h, c_w) \in D$ **do**
4. $\quad c'_h = NoisyPt(c_h, \epsilon, u, v, \delta_\theta, \delta_r, \mathcal{A})$;*// sanitized point*
5. $\quad D' = D' \cup \{(c'_h, c_w)\}$; *// adding sanitized point*
6. **end-for**
7. **return** D';

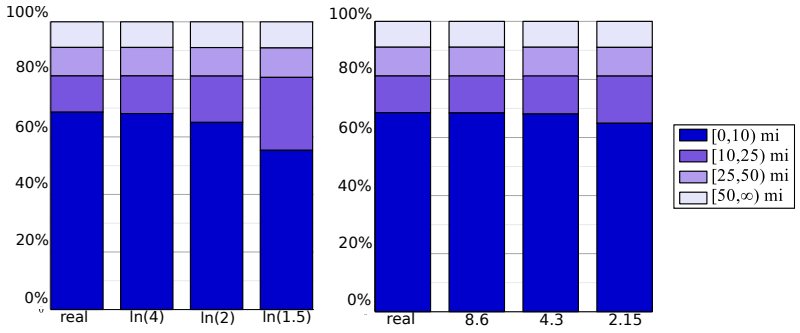
Figure 5.12: Our sanitizing algorithm, based on data perturbation

returns a sanitized counterpart of D. The algorithm is guaranteed to provide ℓ/r-geo-indistinguishability to the home coordinates of all individuals in the dataset (see discussion on protecting multiple locations in Section 5.1.2).

We note that, in contrast to the approach used by the Census Bureau based on histogram's count perturbation, our algorithm modifies the geographical data itself (residence coordinates in this case). Therefore, our algorithm works at a more refined level than the synthetic data generation algorithm used by the Census Bureau; a less refined dataset can be easily obtained however – by just remapping each $(hCoord, wCoord)$ pair produced by our algorithm to its corresponding census block representation.

5.4.1 Experiments on the LODES dataset

In order to evaluate the accuracy of the sanitized dataset generated by our algorithm (and thus of our algorithm as a data sanitizer) we implemented our perturbation mechanism and conducted a series of experiments focusing on the "home-work commute distance" analysis provided by the OnTheMap application. This analysis provides, for a given area (specified as, say, state or county code), a histogram classifying the individuals in the dataset residing in the given area according to the distance between their residence location and their work location. The generated histogram contains four buckets representing different ranges of distance: (1) from



(a) $r = 1.22$ (b) $r = 3.86$

Figure 5.13: Home-work commute distance for various levels ℓ.

zero to ten miles, (2) from ten to twenty five miles, (3) from twenty five to fifty miles, and (4) more than fifty miles.[5].

We have chosen the San Francisco (SF) County as residence area for our experimental analysis. Additionally, we restrict the work location of individuals residing in the San Francisco county to the state of California. The total number of individuals satisfying these conditions amounts to 374.390. All experiments have been carried on using version 6.0 of the LODES dataset. In addition, the mapping from census blocks to their corresponding centroids has been done using the 2011 TIGER census block shapefile information provided by the Census Bureau.

We now proceed to compare the LODES dataset – seen as a histogram – with several sanitized versions of it generated by our algorithm. Figure 5.13 (a) depicts how the geographical information degrades when fixing r to 1.22 miles (so to ensure geo-indistinguishability within 10% of the land area of the SF County) and varying ℓ. The precision parameters were chosen as follows: $u = 10^{-3}$ miles, $\mathcal{A}$'s diameter was set to 10^4 miles, and the standard double precision values for δ_r and δ_θ (for the corresponding ranges).

We have also conducted experiments varying r and fixing ℓ. For instance, if we want to provide geo-indistinguishability for 5%, 10%, and 25% of the land area of the SF county (approx. 46.87 mi^2), we can set $r = 0.86$, 1.22, and 1.93 miles, respectively. Then by taking $\ell = \ln(2)$ we get an histogram very similar to the previous one. This is not surprising as the noise generated by our algorithm depends only on the ratios ℓ/r, which are similar for the values above.

As shown in Figure 5.13 (a), our algorithm has little effect on the bucket counts corresponding to mid/long distance commutes: over twenty five miles the counts of the sanitized dataset are almost identical to those of the input dataset – even for the higher degrees of privacy. For short commutes on the other hand, the increase in privacy degrades the accuracy of the sanitized dataset: several of the commutes that fall in the 0-to-10-miles bucket in the original data fall instead in the 10-to-25-miles bucket in the sanitized data.

[5]Here we choose miles as unit of measure, in order to compare our results with the literature and with online tools about the LODES dataset.

After analyzing the accuracy of the sanitized datasets produced by our algorithm for several levels of privacy, we proceed to compare our approach with the one followed by the Census Bureau to sanitize the LODES dataset. Such comparison is unfortunately not straightforward; on the one hand, the approaches provide different privacy guarantees (see discussion below) and, on the other hand, the Census Bureau is not able to provide us with a (sanitized) dataset sample produced by their algorithm (which would allow us to compare both approaches in terms of accuracy) as this might compromise the protection of the real data.

The algorithm used by the Census Bureau satisfies a notion of privacy called (ϵ, δ)-probabilistic differential privacy (which is a relaxation of standard definition of differential privacy) that provides ϵ-differential privacy with probability at least $1 - \delta$ [Machanavajjhala 2008]. In particular, their algorithm satisfies $(8.6, 0.00001)$ -probabilistic differential privacy. This level of privacy could be compared to geo-indistinguishability for $\ell = 8.6$ and $r = 3.86$, which corresponds to providing protection in an area of the size of the SF County. Figure 5.13 (b) presents the results of our algorithm for such level of privacy and also for higher levels.

It becomes clear that, by allowing high values for ℓ ($\ell = 8.6 = \ln(5432)$, $\ell = 4.3 = \ln(74)$, and $\ell = 2.15 = \ln(9)$) it is possible to provide privacy in large areas without significantly diminishing the quality of the sanitized dataset.

5.5 Comparison with other methods

In this section we compare the performance of our mechanism with that of other ones proposed in the literature. Of course it is not interesting to make a comparison in terms of geo-indistinguishability, since other mechanisms usually do not satisfy this property.

We consider, instead, the (rather natural) Bayesian notion of privacy that measures the expected error of the adversary, presented in Section 3.1. We also consider the trade-off with respect to the quality loss (measured as the expected distance between the real location and the reported result), and also with respect to the notion of accuracy illustrated in the previous section.

The mechanisms that we compare with ours are:

1. The obfuscation mechanism presented in [Shokri 2012]. This mechanism works on discrete locations, called *regions*, and, like ours, it reports a location (region) selected randomly according to a probability distribution that depends on the user's location. The distributions are generated automatically by a tool which is designed to produce an OptPriv mechanism (see Section 3.1), that is, a mechanism that provides optimal privacy for a given utility and a given adversary (i.e., a given prior, representing the side knowledge of the adversary). It is important to note that in presence of a different adversary the optimality is not guaranteed. This dependency on the prior is a key difference with respect to our approach, which abstracts from the adversary's side information.

2. A simple cloaking mechanism. In this approach, the area of interest is assumed to be partitioned in *zones*, whose size depends on the level of privacy we want yo achieve. The mechanism then reports the zone in which the exact location is situated. This method satisfies k-anonymity where k is the number of locations within each zone.

In both cases we need to divide the considered area into a finite number of regions, representing the possible locations. We consider for simplicity a grid, and, more precisely, a 9×9 grid consisting of 81 square regions of 100 m of side length. In addition, for the cloaking method, we overlay a grid of $3 \times 3 = 9$ zones. Figure 5.14 illustrates the setting: the regions are the small squares with black borders. In the cloaking method, the zones are the larger squares with blue borders. For instance, any point situated in one of the regions $1, 2, 3, 10, 11, 12, 19, 20$ or 21, would be reported as zone 1. We assume that each zone is represented by the central region. Hence, in the above example, the reported region would be 11.

Measuring privacy and utility As already stated, we will use the metrics for location privacy and for the quality loss proposed in [Shokri 2012] and described in Section 3.1. The expected error of the advesary is used to measure the privacy offered by a mechanism, and it is formally defined as follows:

$$\textsc{AdvError}(K, \pi) = \min_{H} \sum_{x,\hat{x}} \pi_x (KH)_{x\hat{x}} d_2(x, \hat{x})$$

where π is the prior distribution over the locations, k_{xz} gives the probability that the real location x is reported by the mechanism as z, H is called a remapping, where $h_{z\hat{x}}$ represents the probability that the reported region z is remapped into $\hat{x}$ and d_2 is the euclidean distance between locations. As for the utility, we quantify its opposite, the *Quality Loss* (QL), in terms of the expected distance between the reported location and the user's exact location:

$$\mathrm{QL}(K, \pi) = \sum_{x,z} \pi_x k_{xz} d_2(x, z)$$

where π and k_{xz} are as above. We note that actually the definitions of AdvError and QL presented in Section 3.1 depend respectively on the metric used by the adversary to measure the success of her guessing, and on the one used by the user to measure the quality of the obtained results. In this section we assume that both these metrics are equal to the Euclidean distance d_2, and therefore for simplicity we omit them from the list of arguments.

Recall that for the optimal mechanism in [Shokri 2012] QL and AdvError coincide (when the mechanism is used in presence of the same adversary for which it has been designed), i.e. the adversary does not need to make any remapping.

Comparing privacy for a given utility In order to compare the three mechanisms, we set the parameters of each mechanism in such a way that the QL is the same for all of them, and we compare their privacy in terms of AdvError. As already noted, for the OptPriv mechanism generated by the approach of [Shokri 2012]

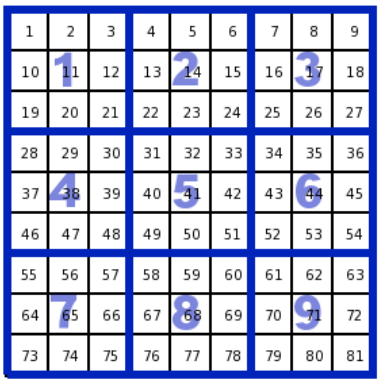


Figure 5.14: The division of the map into regions and zones.

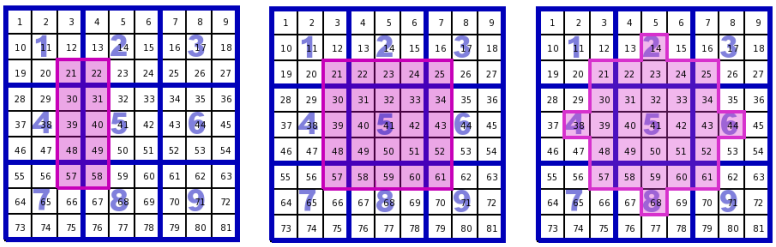


Figure 5.15: Priors considered for the experiments.

QL and AdvError coincide, i.e. the optimal remapping is the identity, when the mechanism is used in presence of the same adversary for which it has been designed. It turns out that, when the adversary's prior is the uniform one, QL and AdvError coincide also for the Planar Laplace mechanism and for the cloaking one.

We note that for the cloaking mechanism the QL is fixed and it is 107.03 m, since there is no randomization involved in the generation of the reported location. Therefore, in our experiments we fix the value of QL to be that one for all the mechanisms. We find that in order to obtain such QL for the Planar Laplace mechanism we need to set $\epsilon = 0.0162$ (the difference with ϵ' in this case is negligible). The OptPriv mechanism of [Shokri 2012] is generated by using the tool explained in the same paper.

Figure 5.15 illustrates the priors π_1, π_2 and π_3 that we consider here: in each case, the probability distribution is accumulated in the regions in the purple area, and distributed uniformly over them. Note that it is not interesting to consider the uniform distribution over the whole map, since, as explained before, on that prior all the mechanisms under consideration give the same result.

Figure 5.16 illustrates the results we obtain when comparing privacy in terms of AdvError, where (a), (b) and (c) correspond to the priors π_1, π_2 and π_3 in Figure 5.15 respectively. The optimal mechanism is considered in two instances: the

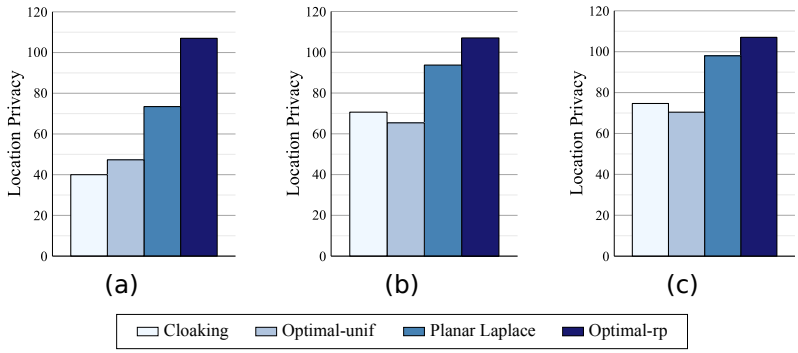


Figure 5.16: Location Privacy, in terms of AdvError, for QL = 107.03 m, for the four mentioned mechanisms, under priors (a) π_1, (b) π_2 and (c) π_3.

one designed exactly for the prior for which it is used (OptPriv_{π_i}, for $i \in \{1, 2, 3\}$), and the one designed for the uniform distribution on all the map (OptPriv_u, which is not necessarily optimal for the priors considered here). As we can see, the Planar Laplace mechanism offers the best location privacy among the mechanisms which do not depend on the prior, or, as in the case of OptPriv_u, are designed with a fixed prior. When the prior has a more circular symmetry the performance approaches the one of the corresponding OptPriv_{π_i} mechanism (which offers the optimal privacy for that prior).

Comparing privacy for a given accuracy The QL metric used above is a reasonable metric, but it does not cover all natural notions of utility. In particular, in the case of LBSs, an important criterion to take into account is the additional bandwidth usage. Therefore, we make now a comparison using the notion of accuracy, which, as explained in previous section, provides a good criterion to evaluate the performance in terms of bandwidth. Unfortunately we cannot compare our mechanism to the one of [Shokri 2012] under this criterion, because the construction of the latter is tied to the a fixed value of QL. Hence, we only compare our mechanism with the cloaking one.

We recall that an LBS application (K, rad_R) is (c, rad_I)-accurate if for every location x the probability that the area of interest (AOI) is fully contained in the area of retrieval (AOR) is at least c. We need to fix rad_I (the radius of the AOI), rad_R (the radius of the AOR), and c so that the condition of accuracy is satisfied for both methods, and then compute the respective AdvError. Let us fix $rad_I = 200$ m, and let us choose a large confidence factor, say, $c = 0.99$. As for rad_R, it will be determined by the cloaking method.

Since the cloaking mechanism is deterministic, in order for the condition to be satisfied the AOR for a given location x must extend around the zone of x by at least rad_I, In fact, x could be in the border of the zone. Given that the cloaking method reports the center of the zone, and that the distance between the center

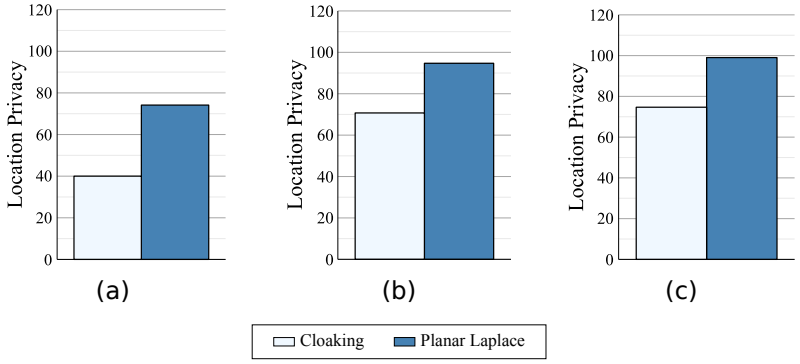


Figure 5.17: Location Privacy for $rad_R = (\sqrt{2} \cdot 150 + 200)$ m and $c = 0.99$.

and the border (which is equal to the distance between the center and any of the corners) is $\sqrt{2} \cdot 150$ m, we derive that rad_R must be at least $(200 + \sqrt{2} \cdot 150)$ m. Note that in the case of this method the accuracy is independent from the value of c. It only depends on the difference between rad_R and rad_I, which in turns depends on the length s of the side of the region: if the difference is at least $\sqrt{2} \cdot s/2$, then the condition is satisfied (for every possible x) with probability 1. Otherwise, there will be some x for which the condition is not satisfied (i.e., it is satisfied with probability 0).

In the case of our method, on the other hand, the accuracy condition depends on c and on ϵ. More precisely, as we have seen in previous section, the condition is satisfied if and only if $C_\epsilon^{-1}(c) \leq rad_R - rad_I$. Therefore, for fixed c, the maximum ϵ only depends on the difference between rad_R and rad_I and is determined by the equation $C_\epsilon^{-1}(c) = rad_R - rad_I$. For the above values of rad_I, rad_R, and c, it turns out that $\epsilon = 0.016$.

We can now compare the AdvError of the two mechanisms with respect to the three priors above. Figure 5.17 illustrates the results. As we can see, our mechanism outperforms the cloaking mechanism in all the three cases.

For different values of rad_I the situation does not change: as explained above, the cloaking method always forces rad_R to be larger than rad_I by (at least) $\sqrt{2} \cdot 150$ m, and ϵ only depends on this value. For smaller values of c, on the contrary, the situation changes, and becomes more favorable for our method. In fact, as argued above, the situation remains the same for the cloaking method (since its accuracy does not depend on c), while ϵ decreases (and consequently AdvError increases) as c decreases. In fact, for a fixed $r = rad_R - rad_I$, we have $\epsilon = C_r^{-1}(c)$. This follows from $r = C_\epsilon^{-1}(c)$ and from the fact that r and ϵ, in the expression that defines $C_\epsilon(r)$, are interchangeable.

5.6 Concluding remarks

Related Work

Much of the related work has been already discussed in Chapter 3, here we only mention the works that were not reported there. There are excellent works and surveys [Terrovitis 2011, Krumm 2009, Shin 2012] that summarize the different threats, methods, and guarantees in the context of location privacy.

LISA [Chen 2009] provides location privacy by preventing an attacker from relating any particular point of interest (POI) to the user's location. That way, the attacker cannot infer which POI the user will visit next. The privacy metric used in this work is *m-unobservability*. The method achieves m-unobservability if, with high probability, the attacker cannot relate the estimated location to at least m different POIs in the proximity. This method does not take into account the attacker's prior knowledge, and it also requires to maintain a database of POIs in the mobile device in order to provide the required privacy.

SpaceTwist [Yiu 2008] reports a fake location (called the "anchor") and queries the geolocation system server incrementally for the nearest neighbors of this fake location until the k-nearest neighbors of the real location are obtained.

Collaborative models were also proposed, where privacy is achieved with a peer-to-peer scheme where users avoid querying the service provider whenever they can find the requested information among their peers [Shokri 2014].

There are also some works whose main goal is to provide accurate results for data mining algorithms while preserving location privacy of the user. Gidofalvi et al. [Gidofalvi 2007] use grid-based anonymization, although the privacy guarantees are mainly experimental.

In [Mironov 2012] it has been shown that, due to finite precision and rounding effects of floating-point operations, the standard implementations of the Laplacian mechanism result in an irregular distribution which causes the loss of the property of differential privacy. In [Gazeau 2013] the study has been extended to the planar Laplacian, and to any kind of finite-precision semantics. The same paper proposes a solutions for the truncated version of the planar laplacian, based on a snapping mechanism, which maintains the level of privacy at the cost of introducing an additional amount of noise.

Dealing with location traces

As discussed in Section 5.1, if we need to add noise to a tuple of n points, then doing it by applying an ϵ-geo-indistinguishable mechanism independently to each point in the tuple guarantees $n\epsilon$-geo-indistinguishability. However, it is clear that in this case the level of privacy decreases fast with respect to the number of points in the tuple.

A location trace is a particular case of tuple, in which not only subsequent points are highly correlated, but also the noise to each points must be added dynamically, i.e. we cannot just add noise to the tuple as a whole. Besides, location traces usually

contain a large number of points, making the previously mentioned approach of adding independent noise ineffective.

To deal with this particular scenario, the authors of [Chatzikokolakis 2013b] have proposed a method to add noise to location traces by means of a prediction function. The basic idea is that, whenever a new point is added to the trace and has to be obfuscated, the mechanism first generates a fake point by "predicting" the direction in which the user has moved (i.e. in this step, no randomization is used). This point is then tested to determine if it falls "too far" from the real location. Then, only if this is the case, a new point is generated by means of a geo-indistinguishable mechanism. This way, the authors succeed in obfuscating traces with a high number of locations with a good level of privacy.

Summary

In this chapter we have adapted the general privacy framework presented in Chapter 4 to the case of location-based applications. As result, we derived a novel notion of location privacy, that we call geo-indistinguishability, and a method, based on a bivariate version of the Laplace function, to perturbate the actual location. We have put a strong emphasis in the formal treatment of the privacy guarantees, both in giving a rigorous definition of geo-indistinguishability, and in providing a mathematical proof that our method satisfies such property.

We have illustrated the applicability of our method on a POI-retrieval service, analyzing the tradeoff between the accuracy of the service and the bandwidth overhead for different levels of privacy. Our experiments show that, by using an affordable amount of bandwidth, we can achieve a good level of privacy and, at the same time, a high level of accuracy. Our method can also be used to sanitize datasets containing location information without degrading significantly the quality of the aggregated results that can be obtained from it, as shown in Section 5.4.

The proposed mechanism compares well with other mechanisms in the literature, and in fact, based on the experiments we performed, it can be seen that it outperforms those which do not depend on the prior.

Chapter 6

Optimal Mechanisms for Location Privacy

In the previous chapter we introduced the notion of geo-indistinguishability, an instance of d-privacy suited for location-based systems. This notion has the property of being independent from the adversary's prior knowledge, meaning that no assumptions are made with respect to the kind of information about the user's location that the attacker has. We also presented the Planar Laplace mechanism, a mechanism based on a bivariate Laplace distribution that achieves this definition. However, the utility of this mechanism is not optimal, in the sense that there are other mechanisms that achieve geo-indistinguishability with the same privacy level, but at the same time offer better utility. In this chapter we will study the trade-off between privacy and utility in geo-indistinguishable mechanisms.

We recall from Section 3.1 that [Shokri 2012] developed an approach to obtain a mechanism that optimizes the privacy while guaranteeing a minimum fixed level of utility. The authors accomplish this by expressing the different privacy and utility constraints as a zero-sum Bayesian Stackelberg game, whose solution can be expressed as a linear optimization problem. However, it must be noted that in this case the privacy is measured as the expected error of the attacker (that we will refer to as AdvError), that is the expected distance between the true location and the best guess of the adversary once she knows the location reported to the LBS:

$$\textsc{AdvError}(K, \pi, d) = \min_{H} \sum_{x,\hat{x}} \pi_x (KH)_{x\hat{x}} d(x, \hat{x})$$

This privacy measure assumes that the adversary knows the prior probability distribution on the user's possible locations. The adversary's guess takes into account the information already in her possession (the prior probability), and it is by definition more accurate, in average, than the reported location. We also say that the adversary may *remap* the reported location.

In this chapter, and following the idea of [Shokri 2012], we aim at optimizing the trade-off between privacy and utility when considering the privacy notion of geo-indistinguishability. In particular, we are interested in the following problem: given a privacy requirement in terms of geo-indistinguishability, we are interested in finding a mechanism that achieves this level of privacy while guaranteeing optimal utility. We recall that we actually measure the opposite of the utility, namely the quality loss of the mechanism, which measures the expected distance of the mechanism under a given prior:

$$\mathrm{QL}(K, \pi, d_Q) = \textsc{ExpDist}(K, \pi, d_Q)$$

First, we present a method, based on a linear optimization problem, to obtain a mechanism that is optimal in the sense just described. Then, we present a method to reduce the number of constraints in the linear program from cubic to quadratic, by using an approximation technique based on spanning graphs. We will also see that the mechanisms obtained this way (either with or without using the approximation technique) are also optimal with respect to the AdvError privacy metric, that is, are OptPriv (see Section 3.1). Finally, perform an evaluation of the proposed approach (in terms of privacy, utility and performance) using information from a dataset containing thousands of traces of several users.

6.1 Geo-indistinguishable mechanisms of optimal utility

As discussed before, we aim at obtaining a mechanism that optimizes the tradeoff between privacy (in terms of geo-indistinguishability) and quality loss (in terms the metric QL). Our main goal is, given a set of locations $\mathcal{X}$ with a privacy metric $d_{\mathcal{X}}$ (typically the Euclidean distance), a privacy level ϵ, a user profile π and a quality metric d_Q, to find an $\epsilon d_{\mathcal{X}}$-private mechanism such that its QL is as small as possible.

We start by describing a set of linear constraints that enforce $\epsilon d_{\mathcal{X}}$-privacy, which allows to obtain an optimal mechanism as a linear optimization problem. However, the number of constraints can be large, making the approach computationally demanding as the number of locations increases. As a consequence, we propose an approximate solution that replaces $d_{\mathcal{X}}$ with the metric induced by a spanning graph. We discuss a greedy algorithm to calculate the spanning graph and analyze its running time. We also show that, if the quality and adversary metrics coincide, then the constructed (exact or approximate) mechanisms also provide optimal privacy in terms of AdvError. Finally, we discuss some practical considerations of our approach.

6.1.1 Constructing an optimal mechanism

The constructed mechanism is assumed to have as both input and output a predetermined finite set of locations $\mathcal{X}$. For instance, $\mathcal{X}$ can be constructed by dividing the map in a finite number of regions (of arbitrary size and shape), and selecting in $\mathcal{X}$ a representative location for each region. We also assume a prior π over $\mathcal{X}$, representing the probability of the user being at each location at any given time.

Given a privacy metric $d_{\mathcal{X}}$ (typically the Euclidean distance) and a privacy parameter ϵ, the goal is to construct an $\epsilon d_{\mathcal{X}}$-private mechanism K such that the *service quality loss* with respect to a quality metric d_Q is minimum. This property is formally defined below:

Definition 6.1. *Given a prior π, a privacy metric $d_{\mathcal{X}}$, a privacy parameter ϵ and a quality metric d_Q, a mechanism K is $\epsilon d_{\mathcal{X}}$-*OptQL(π, d_Q) *iff:*

1. *K is $\epsilon d_{\mathcal{X}}$-private, and*

2. *for all mechanisms K', if K' is $\epsilon d_{\mathcal{X}}$-private then*
   $\mathrm{QL}(K, \pi, d_Q) \leq \mathrm{QL}(K', \pi, d_Q)$

Note that $\epsilon d_{\mathcal{X}}$-OptQL optimizes QL given a privacy constraint, while q-OptPriv (Definition 3.1) optimizes privacy, given a QL constraint.

In order for K to be $\epsilon d_{\mathcal{X}}$-private it should satisfy the following constraints:

$$k_{xz} \leq e^{\epsilon d_{\mathcal{X}}(x,x')} k_{x'z} \qquad x, x', z \in \mathcal{X}$$

Hence, we can construct an optimal mechanism by solving a linear optimization problem, minimizing $\mathrm{QL}(K, \pi, d_Q)$ while satisfying $\epsilon d_{\mathcal{X}}$-privacy:

$$\begin{aligned}
\textbf{Minimize:} \quad & \sum_{x,z \in \mathcal{X}} \pi_x k_{xz} d_Q(x, z) & \\
\textbf{Subject to:} \quad & k_{xz} \leq e^{\epsilon d_{\mathcal{X}}(x,x')} k_{x'z} & x, x', z \in \mathcal{X} \\
& \sum_{z \in \mathcal{X}} k_{xz} = 1 & x \in \mathcal{X} \\
& k_{xz} \geq 0 & x, z \in \mathcal{X}
\end{aligned}$$

It is easy to see that the mechanism K generated by the previous optimization problem is $\epsilon d_{\mathcal{X}}$-OptQL(π, d_Q).

6.1.2 A more efficient method using spanners

In the optimization problem of the previous section, the $\epsilon d_{\mathcal{X}}$-privacy definition introduces $|\mathcal{X}|^3$ constraints in the linear program. However, in order to be able to manage a large number of locations, we would like to reduce this amount to a number in the order of $O(|\mathcal{X}|^2)$. One possible way to achieve this is to use the *dual form* of the linear program. The dual program has as many constraints as the variables of the primal program (in this case $|\mathcal{X}|^2$) and one variable for each constraint in the primal program (in this case $O(|\mathcal{X}|^3)$). Since the primal linear program finds the optimal solution in a finite number of steps, it is guaranteed by the strong duality theorem that dual program will also do so. However, as shown in Section 6.2.3, in practice the dual program does not offer a substantial improvement with respect to the primal one (a possible explanation being that, although fewer in number, the constrains in the dual program are more complex, in the sense that each one of them involves a larger number of variables).

An alternative approach is to exploit the structure of the metric $d_{\mathcal{X}}$. So far we are not making any assumption about $d_{\mathcal{X}}$, and therefore we need to specify $|\mathcal{X}|$ constraints for each pair of locations x and x'. However, it is worth noting that if the distance $d_{\mathcal{X}}$ is induced by a weighted graph (i.e. the distance between each pair of locations is the weight of a minimum path in a graph), then we only need to consider $|\mathcal{X}|$ constraints for each pair of locations that are *adjacent in the graph.* An example of this is the usual definition of differential privacy: since the adjacency relation between databases induces the Hamming distance d_h, we only need to require the

differential privacy constraint for each pair of databases that are adjacent in the Hamming graph (i.e. that differ in one individual).

It might be the case, though, that the metric $d_{\mathcal{X}}$ is not induced by any graph (other than the complete graph), and consequently the amount of constraints remains the same. In fact, this is generally the case for the Euclidean metric. Therefore, we consider the case in which $d_{\mathcal{X}}$ can be *approximated* by some graph-induced metric.

If G is an undirected weighted graph, we denote with d_G the distance function induced by G, i.e. $d_G(x, x')$ denotes the weight of a minimum path between the nodes x and x' in G. Then, if the set of nodes of G is $\mathcal{X}$ and the weight of its edges is given by the metric $d_{\mathcal{X}}$, we can approximate $d_{\mathcal{X}}$ with d_G. In this case, we say that G is a spanning graph, or a spanner [Narasimhan 2007, Sack 1999], of $\mathcal{X}$.

Definition 6.2 (Spanner)**.** *A weighted graph $G = (\mathcal{X}, E)$, with $E \subseteq \mathcal{X} \times \mathcal{X}$ and weight function $w : E \to \mathbb{R}$ is a* spanner *of $\mathcal{X}$ if*

$$w(x, x') = d_{\mathcal{X}}(x, x') \quad \forall (x, x') \in E$$

Note that if G is a spanner of $\mathcal{X}$, then

$$d_G(x, x') \geq d_{\mathcal{X}}(x, x') \quad \forall x, x' \in \mathcal{X}$$

A main concept in the theory of spanners is that of dilation, also known as stretch factor:

Definition 6.3 (Dilation)**.** *Let $G = (\mathcal{X}, E)$ be a spanner of $\mathcal{X}$. The* dilation *of G is calculated as:*

$$\delta = \max_{x \neq x' \in \mathcal{X}} \frac{d_G(x, x')}{d_{\mathcal{X}}(x, x')}$$

*A spanner of $\mathcal{X}$ with dilation δ is called a δ-*spanner *of $\mathcal{X}$.*

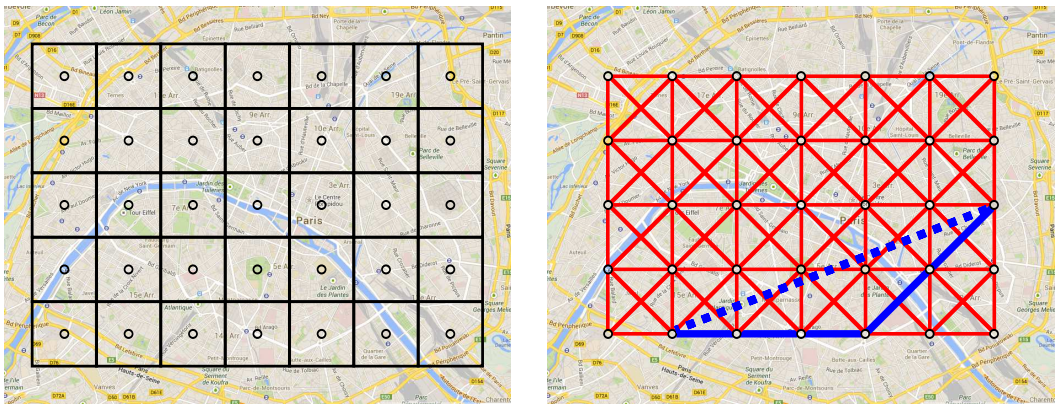
Informally, a δ-spanner of $\mathcal{X}$ can be considered an approximation of the metric $d_{\mathcal{X}}$ in which distances between nodes are "stretched" by a factor of at most δ. Spanners are generally used to approximate distances in a geographic network without considering the individual distances between each pair of nodes. An example of a spanner for a grid in the map can be seen in Figure 6.1.

If G is a δ-spanner of $\mathcal{X}$, then it holds that

$$d_G(x, x') \leq \delta d_{\mathcal{X}}(x, x') \quad \forall x, x' \in \mathcal{X}$$

which leads to the following proposition:

Proposition 6.1. *Let $\mathcal{X}$ be a set of locations with metric $d_{\mathcal{X}}$, and let G be a δ-spanner of $\mathcal{X}$. If a mechanism K for $\mathcal{X}$ is $\frac{\epsilon}{\delta} d_G$-private, then K is $\epsilon d_{\mathcal{X}}$-private.*



(a) (b)

Figure 6.1: (a) a division of the map of Paris into a 7×5 square grid. The set of locations $\mathcal{X}$ contains the centers of the regions. (b) A spanner of $\mathcal{X}$ with dilation $\delta = 1.08$.

We can then propose a new optimization problem to obtain a $\epsilon d_{\mathcal{X}}$-private mechanism. If $G = (\mathcal{X}, E)$ is a δ-spanner of $\mathcal{X}$, we require not the constraints corresponding to $\epsilon d_{\mathcal{X}}$-privacy, but those corresponding to $\frac{\epsilon}{\delta} d_G$-privacy instead, that is, $|\mathcal{X}|$ constraints for each edge of G:

$$\begin{aligned}
\textbf{Minimize:} \quad & \sum_{x,z \in \mathcal{X}} \pi_x k_{xz} d_Q(x,z) & \\
\textbf{Subject to:} \quad & k_{xz} \leq e^{\frac{\epsilon}{\delta} d_G(x,x')} k_{x'z} & z \in \mathcal{X}, (x,x') \in E \\
& \sum_{x \in \mathcal{X}} k_{xz} = 1 & x \in \mathcal{X} \\
& k_{xz} \geq 0 & x, z \in \mathcal{X}
\end{aligned}$$

Since the resulting mechanism is $\frac{\epsilon}{\delta} d_G$-private, by Proposition 6.1 it must also be $\epsilon d_{\mathcal{X}}$-private. However, the number of constraints induced by $\frac{\epsilon}{\delta} d_G$-privacy is now $|E||\mathcal{X}|$. Moreover, as discussed in the next section, for any $\delta > 1$ there is an algorithm that generates a δ-spanner with $O(\frac{|\mathcal{X}|}{\delta - 1})$ edges, which means that, fixing δ, the total number of constraints of the linear program is $O(|\mathcal{X}|^2)$.

It is worth noting that although $\epsilon d_{\mathcal{X}}$-privacy is guaranteed, optimality is lost: the obtained mechanism is $\frac{\epsilon}{\delta} d_G$-OptQL$(\pi, d_Q)$ but not necessarily $\epsilon d_{\mathcal{X}}$-OptQL(π, d_Q), since the set of $\frac{\epsilon}{\delta} d_G$-private mechanisms is a subset of the set of $\epsilon d_{\mathcal{X}}$-private mechanisms. The QL of the obtained mechanism will now depend on the dilation δ of the spanner: the smaller δ is, the closer the QL of the mechanism will be from the optimal one. However, if δ is too small then the number of edges of the spanner will be large, and therefore the number of constraints in the linear program will increase. In fact, when $\delta = 1$ the mechanism obtained is also $\epsilon d_{\mathcal{X}}$-OptQL(π, d_Q) (since d_G and $d_{\mathcal{X}}$ coincide), but the amount of constraints is in general $O(|\mathcal{X}|^3)$. In

consequence, there is a tradeoff between the accuracy of the approximation and the number of constraints in linear program.

6.1.3 An algorithm to construct a δ-spanner

The previous approach requires to compute a spanner for $\mathcal{X}$. Moreover, given a dilation factor δ, we are interested in generating a δ-spanner with a reasonably small number of edges. In this section we describe a simple greedy algorithm to get a δ-spanner of $\mathcal{X}$, presented in [Narasimhan 2007]. This procedure (described in Algorithm 1) is a generalization of Kruskal's minimum spanning tree algorithm.

Algorithm 1 Algorithm to get a δ-spanner of $\mathcal{X}$

```
1:  procedure GetSpanner(X, d_X, δ)
2:      E := ∅
3:      G := (X, E)
4:      for all (x, x') ∈ (X × X) do        ▷ taken in increasing order w.r.t. d_X
5:          if d_G(x, x') > δ d_X(x, x') then
6:              E := E ∪ {(x, x')}
7:          end if
8:      end for
9:      return G
10: end procedure
```

The idea of the algorithm is the following: we start with a spanner with an empty set of edges (lines 2-3). In the main loop we consider all possible edges (that is, all pairs of locations) in *increasing order* with respect to the distance function $d_{\mathcal{X}}$ (lines 4-8), and if the weight of a minimum path between the two corresponding locations in the current graph is bigger than δ times the distance between them, we add the edge to the spanner. By construction, at the end of the procedure, graph G is a δ-spanner of $\mathcal{X}$.

A crucial result presented in [Narasimhan 2007] is that, in the case where $\mathcal{X}$ is a set of points in the Euclidean plane, the degree of each node in the generated spanner only depends on the dilation factor:

Theorem 6.1. *Let $\delta > 1$. If G is a δ-spanner for $\mathcal{X} \subseteq \mathbb{R}^2$, with the Euclidean distance d_2 as metric, then the degree of each node in the spanner constructed by Algorithm 1 is $O(\frac{1}{\delta - 1})$.*

This result is useful to estimate the total number of edges in the spanner, since our goal is to generate a *sparse* spanner, i.e. a spanner with $O(|\mathcal{X}|)$ edges.

Considering the running time of the algorithm, since the main loop requires all pair of regions to be sorted increasingly by distance, we need to perform this sorting before the loop. This step takes $O(|\mathcal{X}|^2 \log |\mathcal{X}|)$. The main loop performs a minimum-path calculation in each step, with $|\mathcal{X}|^2$ total steps. If we use, for instance, Dijkstra's algorithm, each of these operations can be done in $O(|E| +$

$|\mathcal{X}| \log |\mathcal{X}|$). If we select δ so that the final amount of edges in the spanner is linear, i.e. $|E| = O(|\mathcal{X}|)$, we can conclude that the total running time of the main loop is $O(|\mathcal{X}|^3 \log |\mathcal{X}|)$. This turns out to be also the complexity of the whole algorithm.

A common problem in the theory of spanners is the following: given a set of points $\mathcal{X} \subseteq \mathbb{R}^2$ and a maximum amount of edges m, the goal is to find the spanner with *minimum* dilation with at most m edges. This has been proven to be NP-Hard ([Klein 2006]). In our case, we are interested in the analog of this problem: given a maximum tolerable dilation factor δ, we want to find a δ-spanner with minimum amount of edges. However, we can see that the first problem can be expressed in terms of the second (for instance, with a binary search on the dilation factor), which means that the second problems must be at least NP-Hard as well.

6.1.4 AdvError of the obtained mechanism

As discussed in 3.1, the privacy of a location obfuscation mechanism can be expressed in terms of AdvError for an adversary metric d_A. In [Shokri 2012], the problem of optimizing privacy for a given QL constraint is studied, providing a method to obtain a q-OptPriv(π, d_A, d_Q) mechanism for any q, π, d_Q, d_A.

In our case, we optimize QL for a given privacy constraint, constructing a $\epsilon d_{\mathcal{X}}$-OptQL(π, d_Q) mechanism. We now show that, if d_Q and d_A coincide, the mechanism generated by any of the two optimization problems of the previous sections is also q-OptPriv(π, d_Q, d_Q).

AdvError corresponds to an adversary's remapping H that minimizes his expected error with respect to the metric d_A and his prior knowledge π. A crucial observation is that $d_{\mathcal{X}}$-privacy is closed under remapping.

Lemma 6.1. *Let K be a $d_{\mathcal{X}}$-private mechanism, and let H be a remapping. Then KH is $d_{\mathcal{X}}$-private.*

Proof. We know that

$$(KH)_{x\hat{x}} = \sum_{z \in \mathcal{X}} k_{xz} h_{z\hat{x}}, \quad \forall x, \hat{x} \in \mathcal{X}$$

Since K is $\epsilon d_{\mathcal{X}}$-private, we also know that

$$k_{xz} \leq e^{\epsilon d_{\mathcal{X}}(x,x')} k_{x'z}, \quad \forall x, x', z \in \mathcal{X}$$

Therefore, given $x, x' \in \mathcal{X}$, it holds that for all $\hat{x} \in \mathcal{X}$:

$$\begin{aligned}
(KH)_{x\hat{x}} &= \sum_{z \in \mathcal{X}} k_{xz} h_{z\hat{x}} \\
&\leq \sum_{z \in \mathcal{X}} e^{\epsilon d_{\mathcal{X}}(x,x')} k_{x'z} h_{z\hat{x}} \\
&= e^{\epsilon d_{\mathcal{X}}(x,x')} \sum_{z \in \mathcal{X}} k_{x'z} h_{z\hat{x}} \\
&= e^{\epsilon d_{\mathcal{X}}(x,x')} (KH)_{x'\hat{x}}
\end{aligned}$$

and therefore KH is $\epsilon d_{\mathcal{X}}$-private. □

Now let K be a $d_{\mathcal{X}}$-OptQL(π, d_Q) mechanism and H a remapping. Since KH is $d_{\mathcal{X}}$-private (Lemma 6.1) and K is optimal among all such mechanisms, we have that:

$$\mathrm{QL}(K, \pi, d_Q) \leq \mathrm{QL}(KH, \pi, d_Q) \quad \forall H$$

As a consequence, assuming that d_Q and d_A coincide, the adversary minimizes his expected error by applying no remapping at all (i.e. the identity remapping), which means that AdvError$(K, \pi, d_Q) = \mathrm{QL}(K, \pi, d_Q)$ and therefore K must be q-OptPriv(π, d_Q, d_Q).

Theorem 6.2. *If a mechanism K is $d_{\mathcal{X}}$-OptQL(π, d_Q) then it is also q-OptPriv(π, d_Q, d_Q) for $q = \mathrm{QL}(K, \pi, d_Q)$.*

Proof. Let $d_A = d_Q$. We recall from Section 3.1 that for an arbitrary mechanism M, it holds that

$$\begin{aligned} \textsc{AdvError}(M, \pi, d_Q) &= \min_H \textsc{ExpDist}(MH, \pi, d_Q) \\ &= \min_H \mathrm{QL}(MH, \pi, d_Q) \end{aligned}$$

which means that

$$\textsc{AdvError}(M, \pi, d_Q) \leq \mathrm{QL}(M, \pi, d_Q) \tag{1}$$

Let K be a $d_{\mathcal{X}}$-OptQL(π, d_Q) mechanism. Suppose that

$$\textsc{AdvError}(K, \pi, d_Q) < \mathrm{QL}(K, \pi, d_Q)$$

This means that there is a remaping H, other than the identity, such that

$$\mathrm{QL}(KH, \pi, d_Q) \ < \mathrm{QL}(K, \pi, d_Q)$$

However, by Lemma 6.1 we know that KH is also $d_{\mathcal{X}}$-private, and therefore, recalling Definition 6.1, K would not be $d_{\mathcal{X}}$-OptQL(π, d_Q), which is a contradiction. Therefore, we can state that

$$\textsc{AdvError}(K, \pi, d_Q) = \mathrm{QL}(K, \pi, d_Q) \tag{2}$$

Now, in order to see that K is also q-OptPriv(π, d_Q, d_Q), with $q = \mathrm{QL}(K, \pi, d_Q)$, let K' be such that

$$\mathrm{QL}(K', \pi, d_Q) \ \leq \mathrm{QL}(K, \pi, d_Q) \tag{3}$$

According to Definition 3.1 we need to prove that

$$\textsc{AdvError}(K', \pi, d_Q) \leq \textsc{AdvError}(K, \pi, d_Q)$$

And in fact we can see that

$$\begin{aligned} \textsc{AdvError}(K', \pi, d_Q) &\leq \mathrm{QL}(K', \pi, d_Q) && \text{(by (1))} \\ &\leq \mathrm{QL}(K, \pi, d_Q) && \text{(by (3))} \\ &= \textsc{AdvError}(K, \pi, d_Q) && \text{(by (2))} \end{aligned}$$

which concludes our proof.

□

It is important to note that Theorem 6.2 holds for any metric $d_{\mathcal{X}}$. This means that both mechanisms obtained as result of the optimization problems presented in Sections 6.1.1 and 6.1.2 are q-$\textsc{OptPriv}(\pi, d_Q, d_Q)$ – since they are $\epsilon d_{\mathcal{X}}$-$\textsc{OptQL}(\pi, d_Q)$ and $\frac{\epsilon}{\delta} d_G$-$\textsc{OptQL}(\pi, d_Q)$ respectively – however for a different value of q. In fact, in contrast to the method of [Shokri 2012] in which the quality bound q is given as a parameter, our method optimizes the QL given a privacy bound. Hence, the resulting mechanism will be q-$\textsc{OptPriv}(\pi, d_Q, d_Q)$, but for a q that is not known in advance and will depend on the privacy constraint ϵ and the dilation factor δ. The greater the ϵ is (i.e. the higher the privacy), or the lower the δ is (i.e. the better the approximation), the lower the quality loss q of the obtained mechanism will be.

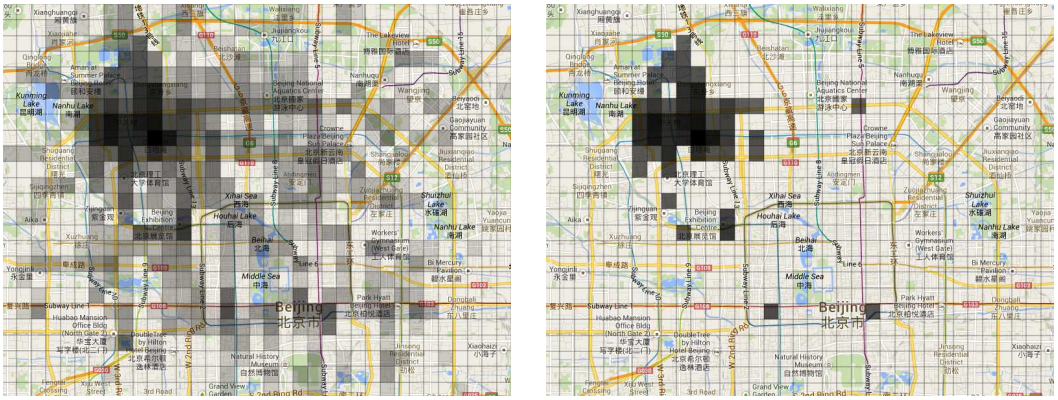
Finally, we must remark that this result only holds in the case where the metrics d_Q, d_A coincide. If the metrics differ, e.g. the quality is measured in terms of the Euclidean distance (the user is interested in accuracy) but the adversary uses the binary distance (he is only interested in the exact location), then this property will no longer be true.

6.1.5 Practical considerations

We conclude this section with a discussion on the practical applicability of location obfuscation. First, it should be noted that, although constructing an optimal mechanism is computationally demanding, once the matrix K is computed, obfuscating a location x only involves drawing a reported location from the distribution $K(x)$ which is computationally trivial. Moreover, although obfuscation is meant to happen on the user's smartphone, computing the mechanism can be offloaded to an external server and even parallelized. The user only needs to transmit $\pi, \epsilon d_{\mathcal{X}}, d_Q$ (which are considered public) and receive K, and the computation only needs to be performed occasionally, to adapt to changes in the user profile.

Second, an important feature of obfuscation mechanisms is that they require no cooperation from the service provider, who simply receives a location and has no way of knowing whether it is real or not. Obfuscation can happen on the user's device, at the operating system or browser level, which is crucial since the user has strong incentives to apply it while the service provider does not. The user's device could also perform filtering of the results, as described in [Andrés 2013].

Finally, we argue that the common idea that users of LBSs are willing to give up their privacy is misleading: the only alternative offered is not to use the service. The usage of browser extensions such as "Location Guard" [Location Guard] shows that



(a) (b)

Figure 6.2: (a) Division of the map of Beijing into regions of size 0.658 x 0.712 km. The density of each region represents its "score", that is, how frequently users visit it. (b) The 50 selected regions. These regions are the ones with highest density between the whole set of regions.

users do care about their privacy and that obfuscation can be a practical approach for using existing services in a privacy friendly way.

6.2 Evaluation

In this section we evaluate the technique for constructing optimal mechanisms described in the previous sections. We perform two kinds of evaluation: first, a comparison with other mechanisms, namely the one of Shokri et al. and the Plannar Laplace mechanism. Second, a performance evaluation of the spanner approximation technique.

The comparison with other mechanisms is performed with respect to both privacy and quality loss. For privacy, the main motivation is to evaluate the mechanisms' privacy under different priors, and in particular under priors different than the one they were constructed with. Following the motivating scenario of the introduction, we consider that a user's profile can vary substantially between different time periods of the day, and simply by taking into account the time of a query, the adversary can obtain a much more informative prior which leads to a lower privacy. For the purposes of the evaluation, we consider priors corresponding to four different time periods: the full day, the morning (7am to noon), afternoon (noon to 7pm) and night (7pm to 7am). Then we construct the mechanisms using the full day prior and compare their privacy for all time periods.

We perform our evaluation on two widely used datasets: GeoLife [Zheng 2008, Zheng 2009, Zheng 2010] and T-Drive [Yuan 2011, Yuan 2010]. The results of the GeoLife dataset are presented in detail in the sections 6.2.1, 6.2.2 and 6.2.3, while

those of the Tdrive dataset (which are in general similar) are summarized in Section 6.2.4.

6.2.1 The GeoLife dataset

The GeoLife GPS Trajectories dataset contains 17621 traces from 182 users, moving mainly in the north-west of Beijing, China, in a period of over five years (from April 2007 to August 2012). The traces show users performing routinary tasks (like going to and from work), and also traveling, shopping, and doing other kinds of entertainment or unusual activities. Besides, the traces were logged by users using different means of transportation, like walking, public transport or bike. More than 90% of the traces were logged in a dense representation, meaning that the individual points in the trace were reported every 1-5 seconds or every 5-10 meters. Since user behaviour changes over time, and the mechanism should be occasionally reconstructed, we restrict each user's traces to a 90 days period, and in particular to the one with the greatest number of recorded traces, so that the prior is as informative as possible.

For the evaluation, we divide the map of Beijing into a grid of regions 0.658 km wide and 0.712 km high, displayed in Figure 6.2a. To avoid users for which little information is available, we only keep those having at least 20 recorded points within the grid area for each one of the time periods. Whenever we count points, those falling within the same grid region during the same hour are counted only once, to prevent traces with a huge number of points in the same region (e.g. the user's home) from completely skewing the results. After this filtering, we end up with 116 users (64% of the total 182).

We then proceed to calculate the 50 "most popular" regions of the grid as follows: for each user, we select the 30 regions in which he spends the greatest amount of time. A region's "score" is the number of users that have it in their 30 highest ranked ones. Then we select the 50 regions with the highest score.

Figure 6.2a shows the division of the map into regions, with the opacity representing the score of each of them, while Figure 6.2b shows the 50 regions with highest score. We can see that most of the selected regions are located in the south-east of the Haidian district, and all of them are located in the north-west of Beijing. We consider the set of locations $\mathcal{X}$ to be the centers of the selected regions, and the metric $d_{\mathcal{X}}$ to be the Euclidean distance between these centers, i.e. $d_{\mathcal{X}} = d_2$.

Finally, a second filtering is performed, again keeping users with at least 20 points in each time period, but this time considering only the 50 selected regions. After this, we end up with a final set of 86 users (46% of the total 182).

6.2.2 Mechanism comparison w.r.t. privacy and quality loss

In this section, we evaluate the location privacy and the utility of three different mechanisms under the several prior distributions for each user. These priors correspond to different parts of the day (all day, morning, afternoon and night), and are

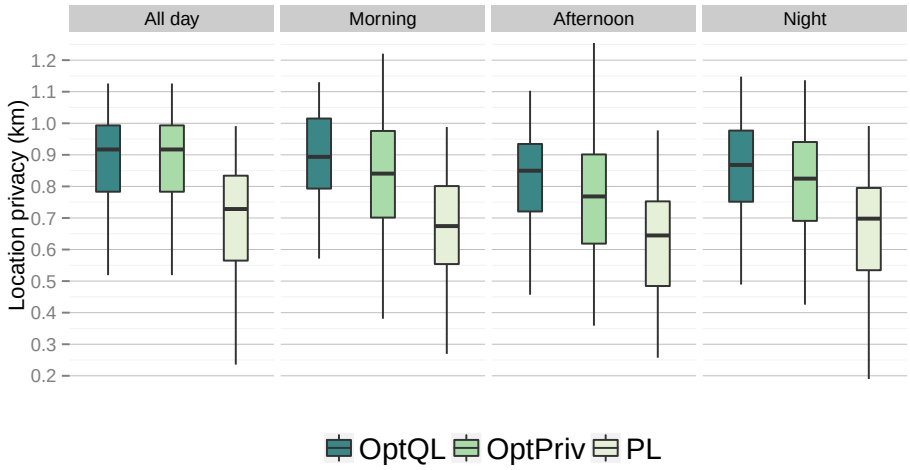


Figure 6.3: Boxplot of the location privacy provided by the three different mechanisms under considered priors. The OptQL mechanism was constructed with $\epsilon = 1.07$ and $\delta = 1.05$.

computed by counting the number of points, logged in the corresponding time period, that fall in each of the selected regions (again, counting only once those points logged within the same hour), and then by normalizing these numbers to obtain a probability distribution.

We start by evaluating the location privacy provided by the different mechanisms. However, we must note that in general location privacy mechanisms do not satisfy $\epsilon d_{\mathcal{X}}$-privacy unless they are specifically designed to do so. Therefore, for this evaluation, we measure location privacy with the metric AdvError, proposed in [Shokri 2012] and described in Section 3.1, which measures the expected error of the attacker under a given prior distribution. In order to perform a fair comparison, we construct the mechanisms in such a way that their QL coincide. The first step is to select a privacy level ϵ and a dilation δ, and then to construct the mechanism described in Section 6.1.2. We will call this mechanism OptQL. This mechanism has a QL of $q = \mathrm{QL}(\textsc{OptQL}, \pi, d_2)$. We then continue by constructing the optimal mechanism of Shokri et al [Shokri 2012], and setting the QL as q. We call this mechanism OptPriv. Finally, we compute a discretized version of the Planar Laplace mechanism of Andrés et al [Andrés 2013]. under a privacy constraint ϵ' (in general different from ϵ) such that the QL of this mechanism is also q. We call this mechanism PL. Note that at the end of this process, by construction, the QL of the three mechanisms is q.

We begin the evaluation comparing the location privacy of each mechanism for each of the selected users, under the four constructed priors. We fix $\epsilon = 1.07$ (which intuitively corresponds to a ratio of 2 between the probability for two regions adjacent in the grid to report the same observed location) and $\delta = 1.05$. Figure 6.3 shows a boxplot of the location privacy (in km) offered by the different mechanisms

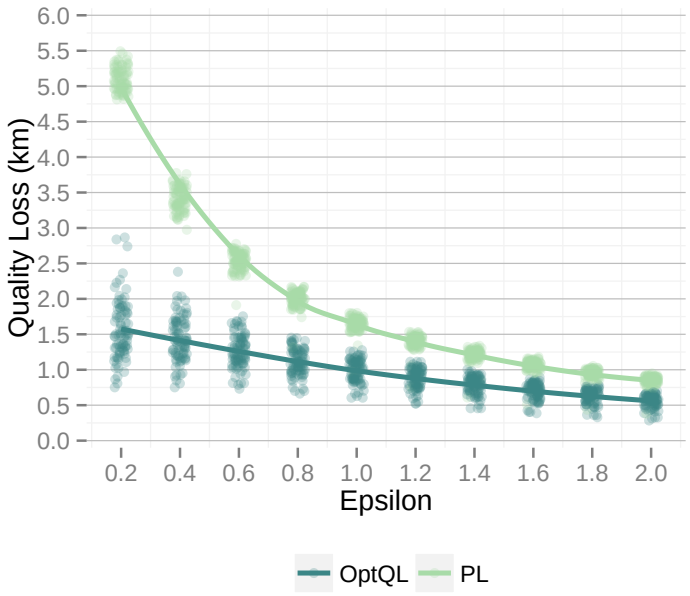


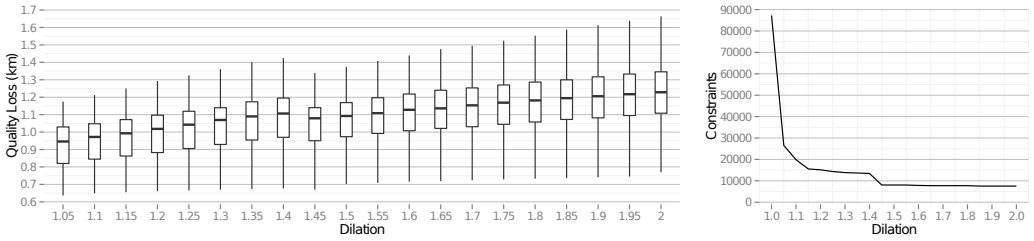
Figure 6.4: Quality loss of the OptQL and PL mechanisms for different values of ϵ. The mechanisms were calculated for all users. Here, points represent the utility for every user, while the two lines join the medians for each mechanism and each value of ϵ.

under each prior. In all four cases, the general performance of our mechanism is better than that of the others, with the only exception being the all-day prior (which is the one used in the construction of the mechanisms) since, as explained in Section 6.1.4, OptQL and OptPriv are q-OptPriv(π, d_2, d_2) and therefore offer the same privacy.

Finally, to show the benefits of using a mechanism with optimal utility, we compare now the QL of the mechanisms OptQL and PL when both mechanisms are generated with the same privacy level ϵ. We can see the results in Figure 6.4. The OptQL mechanism clearly offers a better utility to the user, while guaranteeing the same level of geo-indistinguishability.

6.2.3 Performance of the approximation algorithm

We recall from Section 6.1.2 that if we consider a large number of locations in $\mathcal{X}$, then the number of constraints in the linear program might be large. Hence, we introduced a method based on a spanning graph G to reduce the total number of constraints of the linear program. However, in general the obtained mechanism is no longer $\epsilon d_{\mathcal{X}}$-OptQL(π, d_Q), and therefore it has a higher QL than the optimal one.



(a) (b)

Figure 6.5: (a) Boxplot of the relation between QL and dilation for the mechanism OptQL with privacy constraint $\epsilon = 1.07$. The spanner is calculated with the greedy algorithm presented in Section 6.1.3. (b) Relation between the approximation ratio and the number of constraints in the linear program. This number is independent from the user and form the value of ϵ.

In this section we study the tradeoff between the increase in the QL of the mechanism and the reduction in the number of constraints of the optimization problem, as a consequence of using our approximation technique. We also show how this reduction affects the running time of the whole approach. We start by constructing the OptQL mechanism for all selected users and for different dilations in the range from 1.05 to 2.0, in all cases considering $\epsilon = 1.07$ as before. We then measure the QL of each mechanism under the user profile. We can see the results in Figure 6.5a. It is clear that the QL increases slowly with respect to the dilation: the median value is 0.946 km for $\delta = 1.05$, is 0.972 km for $\delta = 1.1$, and 1.018 km for $\delta = 1.2$. Therefore we can deduce that, for a reasonable approximation, the increase in the quality loss is not really significant. It is worth noting that we do not show the QL for $\delta = 1$ in the plot (corresponding to the case where $d_{\mathcal{X}}$ and d_G are the same). The reason is that in that case the number of constraints is really high, and therefore it takes a lot of time to generate one instance of the mechanism (and much more time to generate it for the 86 users considered).

The relation between the dilation and the number of constraints is shown in Figure 6.5b. Note that this number is independent from the user, and therefore it is enough to calculate it for just one of them. It is clear that the number of constraints decreases exponentially with respect to the dilation, and therefore even for small dilations (which in turn mean good approximations) the number of constraints is significantly reduced with the proposed approximation technique. For instance, we have 87250 constraints for $\delta = 1$ (the optimal case), and 25551 constraints for $\delta = 1.05$. This represents a decrease of 71% with respect to the optimal case, with only 1.05 approximation ratio.

It is also worth noting that, between $\delta = 1.4$ and $\delta = 1.45$ there is a pronounced decrease in the number of constraints (Figure 6.5b) and *also* a decrease in the QL (Figure 6.5a). This might seem counterintuitive at first, since one would expect that a worse approximation should always imply a higher loss of quality. However, there

		Primal simplex		Dual simplex		Interior
$\|\mathcal{X}\|$	δ	Pr. LP	Du. LP	Pr. LP	Du. LP	Pr. LP
50	1.0	57s	1h+	40s	45s	49m 20s
	1.1	46.4s	5.2	5.9s	15.5s	7.5s
	1.2	4m 37s	2s	4s	1h+	2.7s
	1.5	2s	1s	2s	3s	0.5s
	2.0	Error	1s	2s	2s	0.5s
75	1.0	1h+	1h+	29m 26s	1h+	1h+
	1.1	1h+	Error	1m 12s	2m 19s	55s
	1.2	1h+	Error	42s	48.4s	11.7s
	1.5	1h+	5m 55s	19.2s	1h+	2.2s
	2.0	1h+	21.8s	27.2s	15.5s	1.7s

Table 6.1: Execution times of our approach for 50 and 75 locations, for different values of δ, and using different methods to solve the linear program.

is a simple explanation: although the spanner with $\delta = 1.45$ has a higher worst-case approximation ratio, the average-case ratio is actually better that the one of the spanner with $\delta = 1.4$. This phenomenon is a consequence of the particular topology of the set of locations and to the algorithm used to get the spanner.

Finally, we measure the running time of the method used to generate the OptQL mechanism, under different methods to solve the linear optimization problem. The experiments were performed in a 2.8 GHz Intel Core i7 MacBook Pro with 8 GB of RAM running Mac OS X 10.9.1, and the source code for the method was written in C++, using the routines in the GLPK library for the linear program. We compare the performance of three different methods included in the library: the simplex method in both its primal and dual form, and the primal-dual interior-point method. Besides, we run these methods on both the primal linear program presented in Section 6.1.2 and its dual form. Since the running time depends mainly on the number of locations being considered, in the experiments we focus on just one user of the dataset, and we fix the privacy level as $\epsilon = 1.07$. The results can be seen in Table 6.1. Some fields are marked with "1h+", meaning that the execution took more than one hour, after which it was stopped. Others are marked with "Error", meaning that the execution stopped before one hour with an error[1]. A particular case of error happened when running the interior-point method on the dual linear program, where

[1]The actual error message in this case was: "Error: unable to factorize the basis matrix (1). Sorry, basis recovery procedure not implemented yet"

all executions ended with a "numerical instability" error (and therefore this case is not included in the table). From the results we can observe that:

- The only two methods that behave consistently (that never finish with error, and the running time increases when the dilation decreases) are the dual simplex and the interior-point methods, both when applied to the primal program.
- From these, the interior-point method performs better in the case of bigger dilation, while it does it much worse for very small ones.
- Somewhat surprisingly, the dual linear program does not offer a significant performance improvement, specially when compared with the interior-point method.

In the case of OptPriv, the mechanism is generated using Matlab's linear program solver (source code kindly provided by the authors of [Shokri 2012]). We generated the mechanism for the same cases, and observed that the running time mainly depends on the number of regions: for 50 regions, the mechanism is generated in approximately 1 minute, while for 75 regions it takes about 11 minutes.

6.2.4 The T-Drive dataset

In order to reaffirm the validity of the proposed approach, we performed the same evaluation in a different dataset: the T-Drive trajectories dataset. This dataset contains traces of 10357 taxis in Beijing, China, during the period of one week. The total distance of the traces in this dataset is about 9 million kilometres, with more than 15 million reported points. The average time between consecutive points in a trace is 177 seconds, and the average distance is 623 meters.

Due to the huge amount of users in this dataset, we started the evaluation process by blindly selecting (using a standard random function) 5% of the total number users (about 532 users out of 10357). We then perform the same steps as described in the previous sections, particularly those described in Section 6.2.2. In Figure 6.6 we can see the comparison of the location privacy for the different mechanisms. We can see that, also for this dataset, the privacy level of OptQL is, in general, as good as the one of OptPriv, and always better than the one of PL. In particular, the median value for OptQL is always higher than the corresponding one for the other mechanisms (again, with the exception of the all day prior, for which we know that these values coincide). We can also see in Figure 6.7 the comparison in terms of utility of the mechanisms OptQL and PL. Again, the quality loss of OptQL is, in all cases, better than the one of PL. This is to be expected, since, from all mechanisms providing a certain geo-indistinguishability, OptQL is the one with optimal utility (or really close to the optimal utility when the approximation is used).

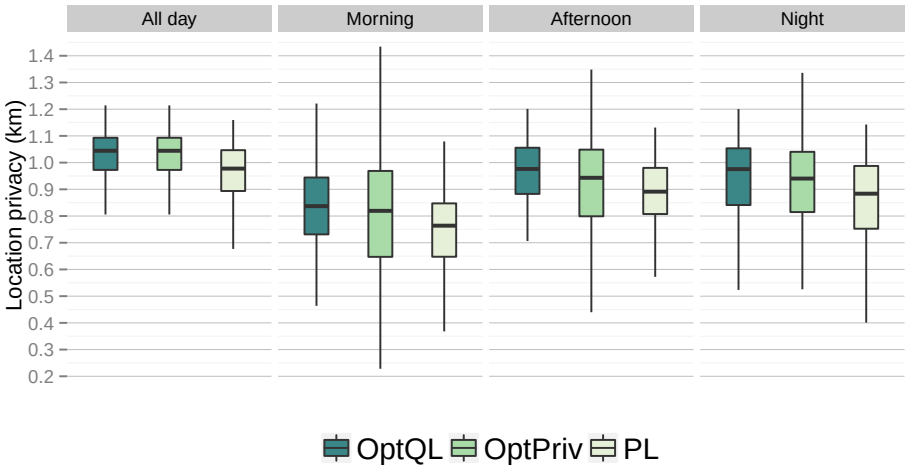


Figure 6.6: Boxplot of the location privacy for the T-Drive dataset. The median value of the location privacy for OptQL is always as good as the one of the other mechanisms.

6.3 Concluding remarks

Related Work

Most of the related work on location privacy has been already discussed in Chapters 3 and 5. However, it is worth mentioning here the work of Herrmann et al. [Herrmann 2013]. The authors present a technique to generate optimal mechanisms under bandwidth and quality constraints. This mechanism are derived by solving linear optimization problems, as in our case. The obfuscation techniques of the obtained mechanisms can be based either on dummy locations, cloaking or simple obfuscation.

Summary

In this chapter we have developed a method to generate a mechanism for location privacy that combines the advantages of the geo-indistinguishability privacy guarantee and the optimal mechanism of [Shokri 2012]. In our approach, a fixed privacy level in terms of geo-indistinguishability is defined in advance, and then a mechanism with optimal utility (one that minimizes the service quality loss for the user) is generated by solving a linear optimization problem. Besides, the privacy guarantee of the obtained mechanism is not affected by the side knowledge of the attacker. Since linear optimization is computationally demanding, we have provided a technique to reduce the total number of constraints in the linear program, based on the use of a spanning graph to approximate distances between locations, which allows a huge reduction on the number of constraints with only a small decrease in the utility. Moreover, in the case where the metric used by the adversary and the one

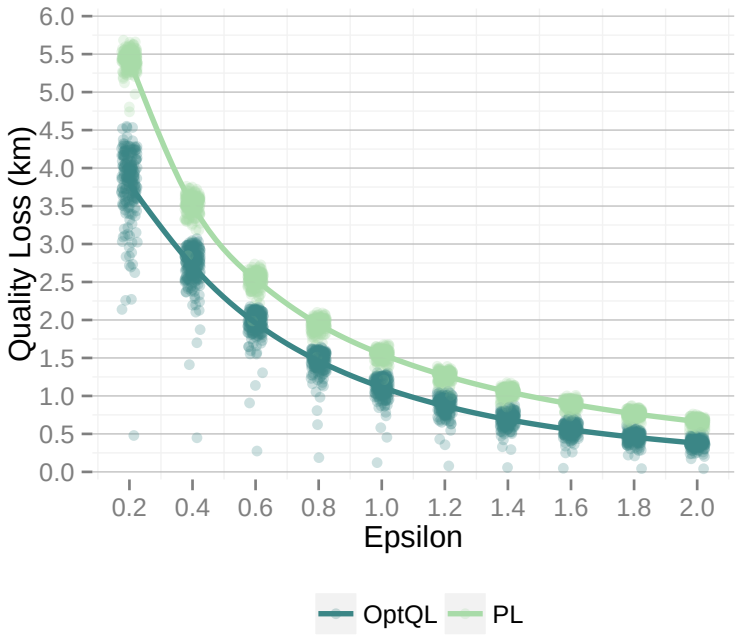


Figure 6.7: Quality loss of the OptQL and PL mechanisms for different values of ϵ, using the data in the T-Drive dataset. The loss of quality of OptQL is always smaller than the one of PL.

used to calculate the utility coincide, then the obtained mechanism is also optimal in the sense described in [Shokri 2012], i.e. when the privacy metric is the expected error of the attacker. Finally, we have evaluated the proposed approach using traces from real users, and we have compared both the privacy and the running time of our mechanism with that of [Shokri 2012]. It turns out that our mechanism offers better privacy guarantees when the side knowledge of the attacker is different from the distribution used to construct the mechanisms. Besides, for a reasonably good approximation factor, we have showed that our approach performs much better in terms of running time.

Chapter 7

Conclusion

In this thesis, we have aimed at developing d-privacy, a general framework to express and measure privacy in a general setting. For this purpose, we extended the well known notion of differential privacy by considering arbitrary domains of secrets (which might differ from the standard case of statistical databases) with different distinguishability metrics. These metrics express the level of distinguishability between secrets, and in general depend not only on the current secret domain, but also in the type of query being considered. We have seen that, by considering different metrics, we are able to capture privacy threats that cannot be captured with the standard notion. At the same time, they allow us to enhance the accuracy of the reported results in queries with high sensitivity in the domain of databases, with respect to standard differential privacy. Besides, being derived from differential privacy, our d-privacy framework inherits the property of being independent from the prior knowledge of the adversary.

We have presented a set of mechanism, based on the Laplace distribution, that achieve d-privacy for different domains of secrets and metrics. We have also seen that, under this framework, the existence of universally optimal mechanisms is not necessarily restricted to counting queries: for metrics other than the usual Hamming distance, universally optimal mechanisms can be found for queries like sum, average and percentile.

We have seen that d-privacy can be successfully applied to domains that do not involve databases. In particular, we derived the novel notion of geo-indistinguishability, a privacy definition in the context of location-based systems, as a particular instance of the d-privacy framework when the set of secrets contains spatial points representing possible locations of an individual. We have also studied the corresponding Laplace mechanism for this scenario, and provided a method to overcome some issues that arise when the mechanism is deployed in real life, namely the privacy loss due to the precision of the machine and the truncation of the reported results into a fixed area of interest. We have shown the applicability of this mechanism in two relevant case studies: the sanitation of a dataset containing location information of several individuals, and the enhancement of a POI-retrieval service with privacy guarantees. Besides, we compared this mechanism with other state-of-the-art approaches, concluding that our mechanism performs better with respect to those mechanisms that do not depend on the prior knowledge of the attacker.

The Laplace mechanism however, might not provide optimal utility in the case of location-based systems. We have studied the trade-off between privacy and utility of geo-indistinguishable mechanisms, and proposed a method, based on linear

optimization, that allows us to retrieve a mechanism that achieves a pre-fixed level of geo-indistinguishability while providing optimal utility for a given user. However, the number of constraints in the linear program is cubic with respect to the number of locations considered. Therefore, we have also proposed a method to approximate distances, based on the use of spanning graphs, and showed that this way we can reduce the number of constraints from cubic to quadratic while maintaining the same level of privacy, with just a little impact on the utility. We have evaluated the proposed approach using traces from a dataset containing information of real users, and compared it with the previously mentioned Laplace mechanism, and also with the optimal mechanism of [Shokri 2012]. We have shown that the optimal mechanism outperforms the other approaches in privacy, offering a significant enhancement in utility with respect to the Laplace mechanism.

Bibliography

[Ács 2011] Gergely Ács and Claude Castelluccia. *I Have a DREAM! (DiffeRentially privatE smArt Metering).* In Tomás Filler, Tomás Pevný, Scott Craver and Andrew D. Ker, editeurs, Proceedings of the 13th International Conference on Information Hiding, (IH 2011), volume 6958 of *Lecture Notes in Computer Science*, pages 118–132. Springer, 2011.

[Andrés 2013] Miguel E. Andrés, Nicolás E. Bordenabe, Konstantinos Chatzikokolakis and Catuscia Palamidessi. *Geo-indistinguishability: differential privacy for location-based systems.* In Proceedings of the 20th ACM Conference on Computer and Communications Security (CCS 2013), pages 901–914. ACM, 2013.

[Ardagna 2007] Claudio Agostino Ardagna, Marco Cremonini, Ernesto Damiani, Sabrina De Capitani di Vimercati and Pierangela Samarati. *Location Privacy Protection Through Obfuscation-Based Techniques.* In Steve Barker and Gail-Joon Ahn, editeurs, Proc. of the 21st Annual IFIP WG 11.3 Working Conference on Data and Applications Security (DAS), volume 4602 of *Lecture Notes in Computer Science*, pages 47–60. Springer, 2007.

[Bamba 2008] Bhuvan Bamba, Ling Liu, Péter Pesti and Ting Wang. *Supporting anonymous location queries in mobile environments with privacygrid.* In Proc. of the 17th International Conference on World Wide Web (WWW), pages 237–246. ACM, 2008.

[Barthe 2012] Gilles Barthe, Boris Köpf, Federico Olmedo and Santiago Zanella Béguelin. *Probabilistic Relational Reasoning for Differential Privacy.* In Proceedings of the 39th Annual ACM Symposium on Principles of Programming Languages (POPL). ACM, 2012.

[Blum 2008] Avrim Blum, Katrina Ligett and Aaron Roth. *A learning theory approach to non-interactive database privacy.* In Proceedings of the 40th Annual ACM Symposium on Theory of Computing (STOC), pages 609–618. ACM, 2008.

[Bordenabe 2014] Nicolás E. Bordenabe, Konstantinos Chatzikokolakis and Catuscia Palamidessi. *Optimal Geo-Indistinguishable Mechanisms for Location Privacy.* In Proceedings of the 21th ACM Conference on Computer and Communications Security (CCS 2014), 2014.

[Brenner 2010] Hai Brenner and Kobbi Nissim. *Impossibility of Differentially Private Universally Optimal Mechanisms.* In Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science (FOCS), pages 71–80. IEEE Computer Society, October 23-26 2010.

[Chatzikokolakis 2013a] Konstantinos Chatzikokolakis, Miguel E. Andrés, Nicolás E. Bordenabe and Catuscia Palamidessi. *Broadening the scope of Differential Privacy using metrics.* In Emiliano De Cristofaro and Matthew Wright, editeurs, Proceedings of the 13th International Symposium on Privacy Enhancing Technologies (PETS 2013), volume 7981 of *Lecture Notes in Computer Science*, pages 82–102. Springer, 2013.

[Chatzikokolakis 2013b] Konstantinos Chatzikokolakis, Catuscia Palamidessi and Marco Stronati. *A Predictive Differentially-Private Mechanism for Mobility Traces.* CoRR, vol. abs/1311.4008, 2013.

[Chen 2009] Zhigang Chen. *Energy-efficient Information Collection and Dissemination in Wireless Sensor Networks.* PhD thesis, University of Michigan, 2009.

[Cheng 2006] Reynold Cheng, Yu Zhang, Elisa Bertino and Sunil Prabhakar. *Preserving User Location Privacy in Mobile Data Management Infrastructures.* In George Danezis and Philippe Golle, editeurs, Proceedings of the 6th International Workshop on Privacy Enhancing Technologies (PET), volume 4258 of *Lecture Notes in Computer Science*, pages 393–412. Springer, 2006.

[Danezis 2011] George Danezis, Markulf Kohlweiss and Alfredo Rial. *Differentially Private Billing with Rebates.* IACR Cryptology ePrint Archive, vol. 2011, page 134, 2011.

[Dewri 2012] Rinku Dewri. *Local Differential Perturbations: Location Privacy Under Approximate Knowledge Attackers.* IEEE Transactions on Mobile Computing, vol. 99, no. PrePrints, page 1, 2012.

[Duckham 2005] Matt Duckham and Lars Kulik. *A Formal Model of Obfuscation and Negotiation for Location Privacy.* In Proc. of the Third International Conference on Pervasive Computing (PERVASIVE), volume 3468 of *Lecture Notes in Computer Science*, pages 152–170. Springer, 2005.

[Dwork 2006a] Cynthia Dwork. *Differential Privacy.* In Michele Bugliesi, Bart Preneel, Vladimiro Sassone and Ingo Wegener, editeurs, 33rd International Colloquium on Automata, Languages and Programming (ICALP 2006), volume 4052 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2006.

[Dwork 2006b] Cynthia Dwork, Frank Mcsherry, Kobbi Nissim and Adam Smith. *Calibrating noise to sensitivity in private data analysis.* In Shai Halevi and Tal Rabin, editeurs, In Proceedings of the Third Theory of Cryptography Conference (TCC), volume 3876 of *Lecture Notes in Computer Science*, pages 265–284. Springer, 2006.

[Dwork 2012] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold and Richard S. Zemel. *Fairness through awareness.* In Shafi Goldwasser, edi-

teur, Proceedings of the Third Innovations in Theoretical Computer Science (ITCS) conference, pages 214–226. ACM, 2012.

[Gambs 2011] Sébastien Gambs, Marc-Olivier Killijian and Miguel Núñez del Prado Cortez. *Show Me How You Move and I Will Tell You Who You Are.* Transactions on Data Privacy, vol. 4, no. 2, pages 103–126, 2011.

[Ganta 2008] Srivatsava Ranjit Ganta, Shiva Prasad Kasiviswanathan and Adam Smith. *Composition attacks and auxiliary information in data privacy.* In Ying Li, Bing Liu and Sunita Sarawagi, editeurs, Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), pages 265–273. ACM, 2008.

[Gazeau 2013] Ivan Gazeau, Dale Miller and Catuscia Palamidessi. *Preserving differential privacy under finite-precision semantics.* In Luca Bortolussi and Herbert Wiklicky, editeurs, Proceedings of the Eleventh Workshop on Quantitative Aspects of Programming Languages (QAPL 2013), volume 117 of *Electronic Proceedings in Theoretical Computer Science*, pages 1–18. Open Publishing Association, 2013.

[Gedik 2005] Bugra Gedik and Ling Liu. *Location Privacy in Mobile Systems: A Personalized Anonymization Model.* In Proc. of the 25th International Conference on Distributed Computing Systems (ICDCS), pages 620–629. IEEE Computer Society, 2005.

[Ghinita 2008] Gabriel Ghinita, Panos Kalnis, Ali Khoshgozaran, Cyrus Shahabi and Kian-Lee Tan. *Private queries in location based services: anonymizers are not necessary.* In Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD), pages 121–132. ACM, 2008.

[Ghosh 2009] Arpita Ghosh, Tim Roughgarden and Mukund Sundararajan. *Universally utility-maximizing privacy mechanisms.* In Proceedings of the 41st annual ACM Symposium on Theory of Computing (STOC), pages 351–360, New York, NY, USA, 2009. ACM.

[Gidofalvi 2007] Gyozo Gidofalvi, Huang Xuegang and Torben Bach Pedersen. *Privacy-Preserving Data Mining on Moving Object Trajectories.* In Proc. of the 8th International Conference on Mobile Data Management (MDM), pages 60 –68, may 2007.

[Google Places] Google Places API. `https://developers.google.com/places/documentation/`.

[Greveler 2012] Ulrich Greveler, Benjamin Justus and Dennis Loehr. *Multimedia Content Identification Through Smart Meter Power Use Profiles.* In 5th International Conference on Computers, Privacy and Data Protection (CPDP 2012), 2012.

[Gruteser 2003] Marco Gruteser and Dirk Grunwald. *Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking.* In Proc. of the First International Conference on Mobile Systems, Applications, and Services (MobiSys). USENIX, 2003.

[Gupte 2010] Mangesh Gupte and Mukund Sundararajan. *Universally optimal privacy mechanisms for minimax agents.* In Jan Paredaens and Dirk Van Gucht, editeurs, Proceedings of the Twenty-Ninth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2010, June 6-11, 2010, Indianapolis, Indiana, USA, pages 135–146. ACM, 2010.

[Herrmann 2013] Michael Herrmann, Carmela Troncoso, Claudia Diaz and Bart Preneel. *Optimal Sporadic Location Privacy Preserving Systems in Presence of Bandwidth Constraints.* In Proceedings of the 2013 ACM Workshop on Privacy in the Electronic Society (WPES), 2013.

[Ho 2011] Shen-Shyang Ho and Shuhua Ruan. *Differential privacy for location pattern mining.* In Proceedings of the 4th ACM SIGSPATIAL International Workshop on Security and Privacy in GIS and LBS (SPRINGL), pages 17–24. ACM, 2011.

[Hoh 2005] Baik Hoh and Marco Gruteser. *Protecting Location Privacy Through Path Confusion.* In Proc. of SecureComm, pages 194–205. IEEE, 2005.

[Khoshgozaran 2007] Ali Khoshgozaran and Cyrus Shahabi. *Blind Evaluation of Nearest Neighbor Queries Using Space Transformation to Preserve Location Privacy.* In Proc. of the 10th International Symposium on the Advances in Spatial and Temporal Databases (SSTD), volume 4605 of *Lecture Notes in Computer Science*, pages 239–257. Springer, 2007.

[Kido 2005] Hidetoshi Kido, Yutaka Yanagisawa and Tetsuji Satoh. *Protection of Location Privacy using Dummies for Location-based Services.* In Proc. of ICDE Workshops, page 1248, 2005.

[Klein 2006] Rolf Klein and Martin Kutz. *Computing Geometric Minimum-Dilation Graphs is NP-Hard.* In Proceedings of the 14th International Symposium on Graph Drawing (GD 2006), volume 4372, pages 196–207. Springer, 2006.

[Krumm 2009] John Krumm. *A survey of computational location privacy.* Personal and Ubiquitous Computing, vol. 13, no. 6, pages 391–399, 2009.

[Lam 2007] H. Lam, G. Fung and W. Lee. *A novel method to construct taxonomy electrical appliances based on load signatures.* IEEE Transactions on Consumer Electronics, vol. 53, no. 4, pages 653—660, 2007.

[Li 2007] Ninghui Li, Tiancheng Li and Suresh Venkatasubramanian. *t-Closeness: Privacy Beyond k-Anonymity and l-Diversity.* In ICDE, volume 7, pages 106–115, 2007.

[Machanavajjhala 2007] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke and Muthuramakrishnan Venkitasubramaniam. *l-diversity: Privacy beyond k-anonymity.* ACM Transactions on Knowledge Discovery from Data (TKDD), vol. 1, no. 1, page 3, 2007.

[Machanavajjhala 2008] Ashwin Machanavajjhala, Daniel Kifer, John M. Abowd, Johannes Gehrke and Lars Vilhuber. *Privacy: Theory meets Practice on the Map.* In Gustavo Alonso, José A. Blakeley and Arbee L. P. Chen, editeurs, Proceedings of the 24th International Conference on Data Engineering, ICDE 2008, April 7-12, 2008, Cancún, México, pages 277–286. IEEE, 2008.

[Pew Internet. Location-Based Services 2013] Pew Internet. Location-Based Services, 2013. `http://www.pewinternet.org/2013/09/12/location-based-services/`.

[Pew Internet. Smartphone Ownership 2013] Pew Internet. Smartphone Ownership, 2013.

[McSherry 2007] Frank McSherry and Kunal Talwar. *Mechanism Design via Differential Privacy.* In Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2007), pages 94–103. IEEE Computer Society, 2007.

[Mironov 2012] Ilya Mironov. *On significance of the least significant bits for differential privacy.* In Ting Yu, George Danezis and Virgil D. Gligor, editeurs, Proceedings of the ACM Conference on Computer and Communications Security (CCS), pages 650–661. ACM, 2012.

[Mokbel 2006] Mohamed F. Mokbel, Chi-Yin Chow and Walid G. Aref. *The New Casper: Query Processing for Location Services without Compromising Privacy.* In Umeshwar Dayal, Kyu-Young Whang, David B. Lomet, Gustavo Alonso, Guy M. Lohman, Martin L. Kersten, Sang Kyun Cha and Young-Kuk Kim, editeurs, Proceedings of the 32nd International Conference on Very Large Data Bases (VLDB), pages 763–774. ACM, 2006.

[Narasimhan 2007] Giri Narasimhan and Michiel Smid. Geometric spanner networks. Cambridge University Press, 2007.

[Narayanan 2008] Arvind Narayanan and Vitaly Shmatikov. *Robust Deanonymization of Large Sparse Datasets.* In Proceedings of the 29th IEEE Symposium on Security and Privacy, pages 111–125, 2008.

[Narayanan 2009] Arvind Narayanan and Vitaly Shmatikov. *De-anonymizing social networks.* In Security and Privacy, 2009 30th IEEE Symposium on, pages 173–187. IEEE, 2009.

[Nissim 2007] Kobbi Nissim, Sofya Raskhodnikova and Adam Smith. *Smooth sensitivity and sampling in private data analysis.* In David S. Johnson and Uriel

Feige, editeurs, Proceedings of the 39th Annual ACM Symposium on Theory of Computing (STOC), pages 75–84. ACM, 2007.

[Reed 2010] Jason Reed and Benjamin C. Pierce. *Distance makes the types grow stronger: a calculus for differential privacy.* In Paul Hudak and Stephanie Weirich, editeurs, Proceeding of the 15th ACM SIGPLAN International Conference on Functional Programming (ICFP), pages 157–168, Baltimore, Maryland, USA, September 27-29 2010. ACM.

[Roth 2010] Aaron Roth and Tim Roughgarden. *Interactive privacy via the median mechanism.* In Proc. of the 42nd ACM Symposium on Theory of Computing (STOC), pages 765–774, 2010.

[Rubin 1993] Donald B. Rubin. *Discussion: Statistical Disclosure Limitation.* Journal of Official Statistics, vol. 9, no. 2, pages 461–468, 1993.

[Sack 1999] J.R. Sack and J. Urrutia. Handbook of computational geometry. Elsevier Science, 1999.

[Samarati 2001] Pierangela Samarati. *Protecting Respondents' Identities in Microdata Release.* IEEE Trans. Knowl. Data Eng, vol. 13, no. 6, pages 1010–1027, 2001.

[Shankar 2009] Pravin Shankar, Vinod Ganapathy and Liviu Iftode. *Privately querying location-based services with SybilQuery.* In Sumi Helal, Hans Gellersen and Sunny Consolvo, editeurs, Proc. of the 11th International Conference on Ubiquitous Computing (UbiComp), pages 31–40. ACM, 2009.

[Shin 2012] Kang G. Shin, Xiaoen Ju, Zhigang Chen and Xin Hu. *Privacy protection for users of location-based services.* IEEE Wireless Commun, vol. 19, no. 2, pages 30–39, 2012.

[Shokri 2010] Reza Shokri, Carmela Troncoso, Claudia Diaz, Julien Freudiger and Jean-Pierre Hubaux. *Unraveling an Old Cloak: k-anonymity for Location Privacy.* In Proceedings of the 9th annual ACM Workshop on Privacy in the Electronic Society (WPES 2010), pages 115–118 115–118 115–118, 2010.

[Shokri 2011] Reza Shokri, George Theodorakopoulos, Jean-Yves Le Boudec and Jean-Pierre Hubaux. *Quantifying Location Privacy.* In IEEE Symposium on Security and Privacy, pages 247–262. IEEE Computer Society, 2011.

[Shokri 2012] Reza Shokri, George Theodorakopoulos, Carmela Troncoso, Jean-Pierre Hubaux and Jean-Yves Le Boudec. *Protecting location privacy: optimal strategy against localization attacks.* In Ting Yu, George Danezis and Virgil D. Gligor, editeurs, Proceedings of the 19th ACM Conference on Computer and Communications Security (CCS 2012), pages 617–627. ACM, 2012.

[Shokri 2014] Reza Shokri, George Theodorakopoulos, Panos Papadimitratos, Ehsan Kazemi and Jean-Pierre Hubaux. *Hiding in the Mobile Crowd: Location Privacy through Collaboration*. In Proceedings of the IEEE Transactions on Dependable and Secure Computing (TDSC 2014). IEEE, 2014.

[Terrovitis 2011] Manolis Terrovitis. *Privacy preservation in the dissemination of location data*. SIGKDD Explorations, vol. 13, no. 1, pages 6–18, 2011.

[Vodafone] Vodafone Mobile data usage Stats. `http://www.vodafone.ie/internet-broadband/internet-on-your-mobile/usage/`.

[Xue 2009] Mingqiang Xue, Panos Kalnis and Hung Pung. *Location Diversity: Enhanced Privacy Protection in Location Based Services*. In Proc. of the 4th International Symposium on Location and Context Awareness (LoCA), volume 5561 of *Lecture Notes in Computer Science*, pages 70–87. Springer, 2009.

[Yiu 2008] Man Lung Yiu, Christian S. Jensen, Xuegang Huang and Hua Lu. *SpaceTwist: Managing the Trade-Offs Among Location Privacy, Query Performance, and Query Accuracy in Mobile Services*. In Gustavo Alonso, José A. Blakeley and Arbee L. P. Chen, editeurs, Proceedings of the 24th International Conference on Data Engineering (ICDE), pages 366–375. IEEE, 2008.

[Yuan 2010] Jing Yuan, Yu Zheng, Chengyang Zhang, Wenlei Xie, Xing Xie, Guangzhong Sun and Yan Huang. *T-drive: driving directions based on taxi trajectories*. In GIS, pages 99–108, 2010.

[Yuan 2011] Jing Yuan, Yu Zheng, Xing Xie and Guangzhong Sun. *Driving with knowledge from the physical world*. In The 17th ACM SIGKDD international conference on Knowledge Discovery and Data mining, KDD '11, 2011.

[Zheng 2008] Yu Zheng, Quannan Li, Yukun Chen, Xing Xie and Wei-Ying Ma. *Understanding Mobility Based on GPS Data*. In Proceedings of ACM conference on Ubiquitous Computing (UbiComp 2008), 2008.

[Zheng 2009] Yu Zheng, Lizhu Zhang, Xing Xie and Wei-Ying Ma. *Mining interesting locations and travel sequences from GPS trajectories*. In Proceedings of International conference on World Wild Web (WWW 2009), 2009.

[Zheng 2010] Yu Zheng, Xing Xie and Wei-Ying Ma. *GeoLife: A Collaborative Social Networking Service among User, Location and Trajectory*. IEEE Data Eng. Bull., vol. 33, no. 2, pages 32–39, 2010.

[Location Guard] Location Guard. `https://github.com/chatziko/location-guard`.